

LAPORAN PRAKTIKUM PEMROGRAMAN BACKEND LANJUTAN
CLEAN ARSITEKTUR, STRUCT TRASH DAN MODUL 8



Disusun Oleh :

Shelyna Riska Amanatullah

434231005

Dosen Pengampu :

Moh. Nasrul Aziz, S.Kom., M.Kom

Anugrah Anang Prasetyo, S.Kom S.Kom

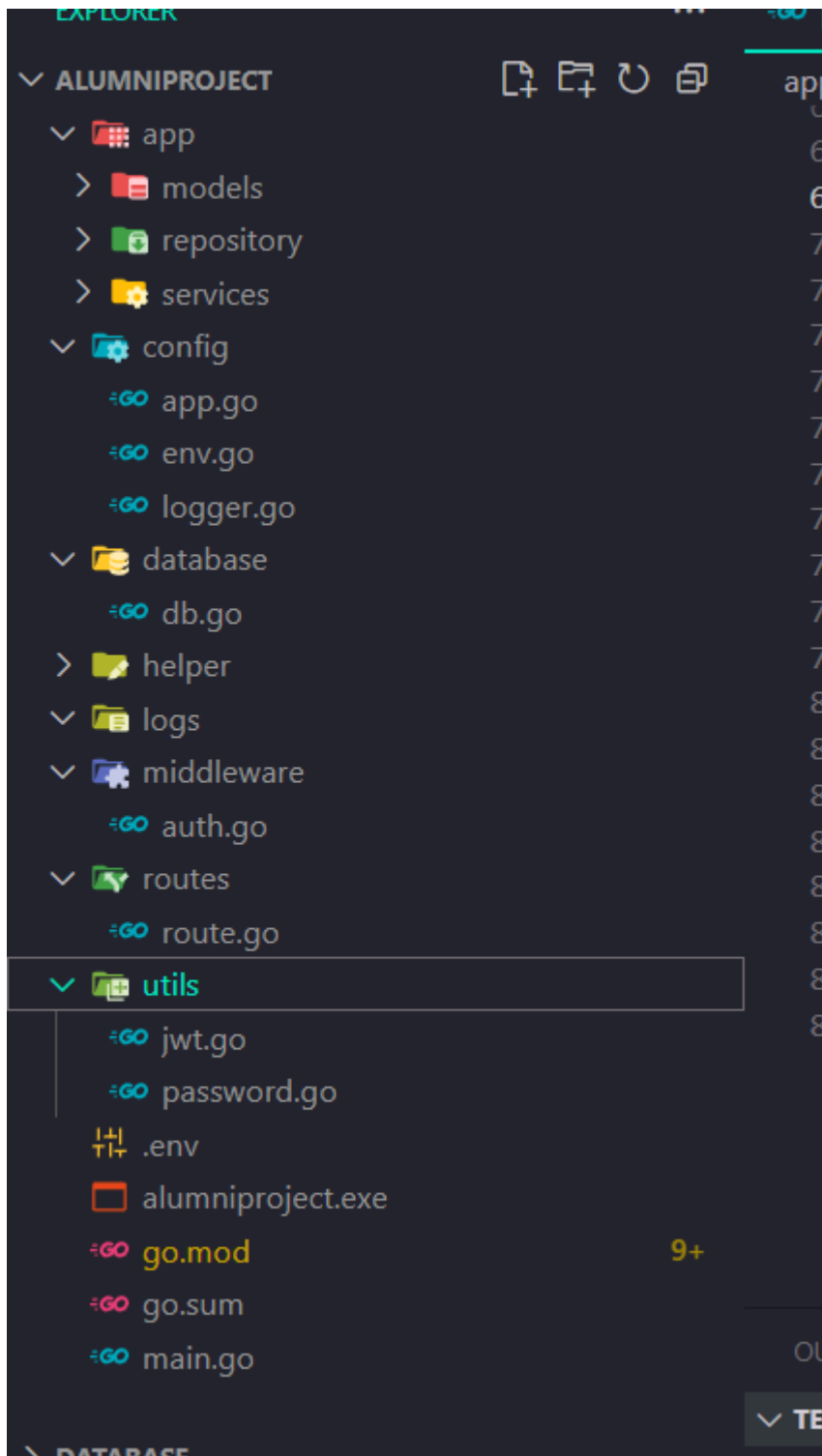
D4 TEKNIK INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS AIRLANGGA

2025

1. Clean Arsitektur



2. Model Trash

```
type TrashPekerjaan struct {  
    ID          int          `json:"id"`  
    NamaPekerjaan string      `json:"nama_pekerjaan"`  
    TempatKerja string      `json:"tempat_kerja"`  
    DeletedAt   *time.Time `json:"deleted_at"`  
}
```

```

}

type GetTrashPekerjaan 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"`
    StatusPekerjaan string       `json:"status_pekerjaan"`
    DeletedAt     *time.Time   `json:"deleted_at"`
    CreatedBy     int           `json:"created_by"`
}

```

3. Repository Trash

```

func GetTrashPekerjaan() ([]models.GetTrashPekerjaan, error) {
    ctx, cancel := context.WithTimeout(context.Background(),
5*time.Second)
    defer cancel()

    query := `
        SELECT id, alumni_id, nama_perusahaan, posisi_jabatan,
bidang_industri,
                lokasi_kerja, status_pekerjaan, deleted_at,
created_by
        FROM pekerjaan_alumni
        WHERE deleted_at IS NOT NULL
        ORDER BY deleted_at DESC;
    `

    rows, err := database.DB.QueryContext(ctx, query)
    if err != nil {
        return nil, err
    }
    defer rows.Close()

    var result []models.GetTrashPekerjaan
    for rows.Next() {
        var t models.GetTrashPekerjaan
        if err := rows.Scan(
            &t.ID,
            &t.AlumniID,
            &t.NamaPerusahaan,

```

```

        &t.PosisiJabatan,
        &t.BidangIndustri,
        &t.LokasiKerja,
        &t.StatusPekerjaan,
        &t.DeletedAt,
        &t.CreatedBy, // pastikan ada di SELECT
    ); err != nil {
        return nil, err
    }
    result = append(result, t)
}

return result, nil
}

```

4. Services Trash

```

func GetTrashPekerjaanService(c *fiber.Ctx) error {
    userRole := c.Locals("role").(string)
    userID := c.Locals("user_id").(int)

    allTrash, err := repository.GetTrashPekerjaan()
    if err != nil {
        return c.Status(500).JSON(fiber.Map{"error": "Gagal mengambil data trash"})
    }

    if userRole == "user" {
        var filtered []models.GetTrashPekerjaan
        for _, p := range allTrash {
            if p.CreatedBy == userID {
                filtered = append(filtered, p)
            }
        }
        return c.JSON(fiber.Map{
            "success": true,
            "count": len(filtered),
            "data": filtered,
        })
    }
    // Admin dapat semua
    return c.JSON(fiber.Map{
        "success": true,

```

```

        "count":    len(allTrash) ,
        "data":     allTrash,
    })
}

```

5. Routes Trash

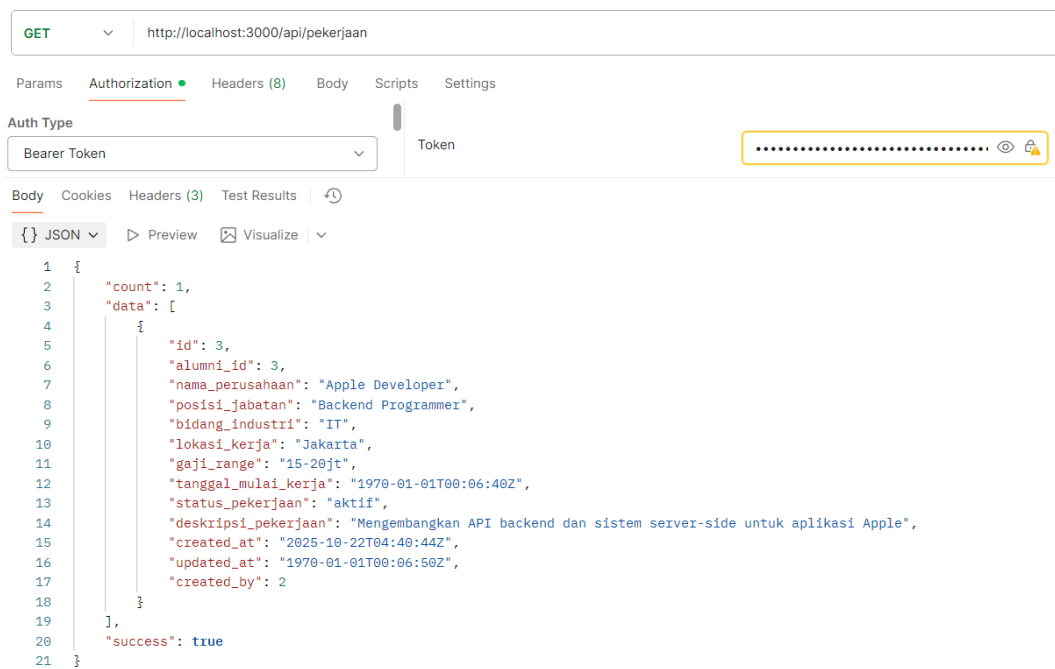
```

// Pekerjaan routes
pekerjaan := protected.Group("/pekerjaan")
pekerjaan.Get("/trash", service.GetTrashPekerjaanService) //

```

6. CRUD Mongodb

a. Get Pekerjaan (User)



GET http://localhost:3000/api/pekerjaan

Auth Type: Bearer Token

Token: [REDACTED]

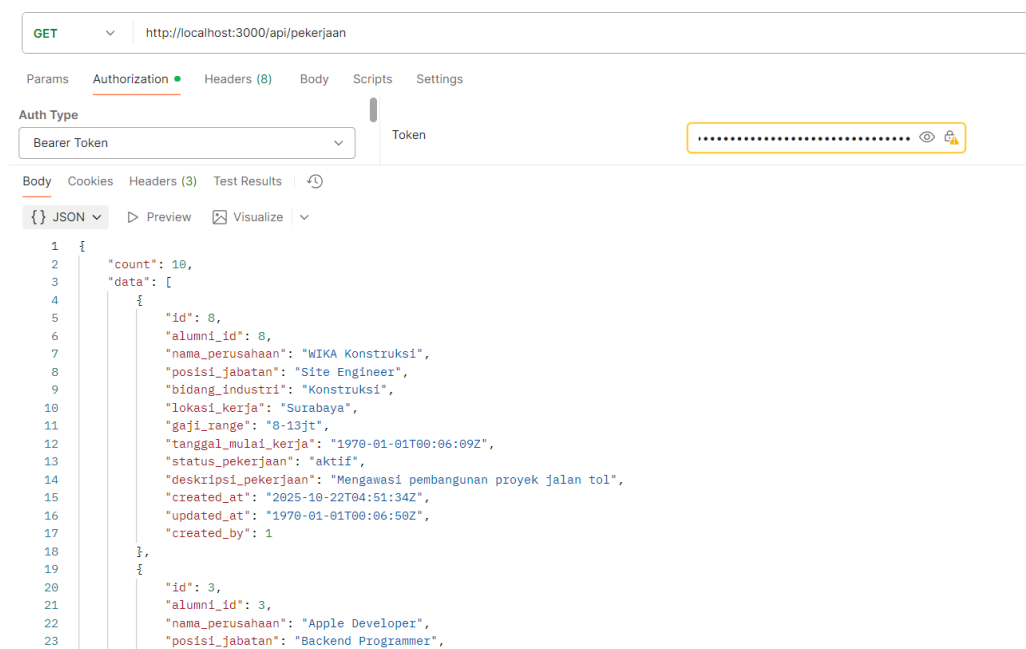
Body: JSON

```

1  {
2      "count": 1,
3      "data": [
4          {
5              "id": 3,
6              "alumni_id": 3,
7              "nama_perusahaan": "Apple Developer",
8              "posisi_jabatan": "Backend Programmer",
9              "bidang_industri": "IT",
10             "lokasi_kerja": "Jakarta",
11             "gaji_range": "15-20jt",
12             "tanggal_mulai_kerja": "1970-01-01T00:06:40Z",
13             "status_pekerjaan": "aktif",
14             "deskripsi_pekerjaan": "Mengembangkan API backend dan sistem server-side untuk aplikasi Apple",
15             "created_at": "2025-10-22T04:40:44Z",
16             "updated_at": "1970-01-01T00:06:50Z",
17             "created_by": 2
18         }
19     ],
20     "success": true
21 }

```

b. Get Pekerjaan (Admin)



GET http://localhost:3000/api/pekerjaan

Auth Type: Bearer Token

Token: [REDACTED]

Body: JSON

```

1  {
2      "count": 10,
3      "data": [
4          {
5              "id": 8,
6              "alumni_id": 8,
7              "nama_perusahaan": "WIKA Konstruksi",
8              "posisi_jabatan": "Site Engineer",
9              "bidang_industri": "Konstruksi",
10             "lokasi_kerja": "Surabaya",
11             "gaji_range": "8-13jt",
12             "tanggal_mulai_kerja": "1970-01-01T00:06:09Z",
13             "status_pekerjaan": "aktif",
14             "deskripsi_pekerjaan": "Mengawasi pembangunan proyek jalan tol",
15             "created_at": "2025-10-22T04:51:34Z",
16             "updated_at": "1970-01-01T00:06:50Z",
17             "created_by": 1
18         },
19         {
20             "id": 3,
21             "alumni_id": 3,
22             "nama_perusahaan": "Apple Developer",
23             "posisi_jabatan": "Backend Programmer",

```

c. Get Pekerjaan/ID (User)

GET http://localhost:3000/api/pekerjaan/3

Params Authorization Headers (8) Body Scripts Settings

Auth Type: Bearer Token

Token: [REDACTED]

Body: Cookies Headers (3) Test Results

JSON Preview Visualize

```
1 {
2   "data": {
3     "id": 3,
4     "alumni_id": 3,
5     "nama_perusahaan": "Apple Developer",
6     "posisi_jabatan": "Backend Programmer",
7     "bidang_industri": "IT",
8     "lokasi_kerja": "Jakarta",
9     "gaji_range": "15-20jt",
10    "tanggal_mulai_kerja": "1970-01-01T00:06:40Z",
11    "status_pekerjaan": "aktif",
12    "deskripsi_pekerjaan": "Mengembangkan API backend dan sistem server-side untuk aplikasi Apple",
13    "created_at": "2025-10-22T04:40:44Z",
14    "updated_at": "1970-01-01T00:06:50Z",
15    "created_by": 2
16  },
17  "message": "Data pekerjaan berhasil diambil",
18  "success": true
19 }
```

d. Get Pekerjaan/ID (Admin)

GET http://localhost:3000/api/pekerjaan/1

Params Authorization Headers (8) Body Scripts Settings

Auth Type: Bearer Token

Token: [REDACTED]

Body: Cookies Headers (3) Test Results

JSON Preview Visualize

```
1 {
2   "data": {
3     "id": 1,
4     "alumni_id": 1,
5     "nama_perusahaan": "PT Maju Jaya",
6     "posisi_jabatan": "Software Engineer",
7     "bidang_industri": "IT",
8     "lokasi_kerja": "Jakarta",
9     "gaji_range": "10-15jt",
10    "tanggal_mulai_kerja": "1970-01-01T00:06:29Z",
11    "tanggal_selesai_kerja": "2025-10-22T04:34:11Z",
12    "status_pekerjaan": "aktif",
13    "deskripsi_pekerjaan": "Fullstack Developer, mengembangkan aplikasi web dan mobile",
14    "created_at": "2025-10-22T04:34:11Z",
15    "updated_at": "1970-01-01T00:06:50Z",
16    "created_by": 1
17  },
18  "message": "Data pekerjaan berhasil diambil",
19  "success": true
20 }
```

e. Post Pekerjaan (Admin)

POST ▼ http://localhost:3000/api/pekerjaan

Params Authorization Headers (10) **Body** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   "_id": 13,
3   "alumni_id": 13,
4   "nama_perusahaan": "Tokopedia",
5   "posisi_jabatan": "Data Scientist",
6   "bidang_industri": "E-Commerce",
7   "lokasi_kerja": "Jakarta",
8   "gaji_range": "10-15jt",
9   "status_pekerjaan": "aktif",
10  "deskripsi_pekerjaan": "Analisis data besar untuk optimasi bisnis",
11  "tanggal_mulai_kerja": "2021-06-01",
12  "tanggal_selesai_kerja": null
13 }
```

Body Cookies Headers (3) Test Results ↻

{} JSON ▼ ▶ Preview 🖼 Visualize ▼

```
2   "data": {
3     "id": 11,
4     "alumni_id": 13,
5     "nama_perusahaan": "Tokopedia",
6     "posisi_jabatan": "Data Scientist",
7     "bidang_industri": "E-Commerce",
8     "lokasi_kerja": "Jakarta",
9     "gaji_range": "10-15jt",
10    "tanggal_mulai_kerja": "2021-06-01T00:00:00Z",
11    "status_pekerjaan": "aktif",
12    "deskripsi_pekerjaan": "Analisis data besar untuk optimasi bisnis",
13    "created_at": "2025-10-22T12:55:21.9083355+07:00"
```

f. Put Pekerjaan (Admin)

PUT ▼ http://localhost:3000/api/pekerjaan/11

Params Authorization Headers (10) **Body** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   "_id": 13,
3   "alumni_id": 13,
4   "nama_perusahaan": "Shopee",
5   "posisi_jabatan": "Data Scientist",
6   "bidang_industri": "E-Commerce",
7   "lokasi_kerja": "Jakarta",
8   "gaji_range": "10-15jt",
9   "status_pekerjaan": "aktif"
10 }
```

Body Cookies Headers (3) Test Results ↻

{} JSON ▼ ▶ Preview 🖼 Visualize ▼

```
1 {
2   "message": "Pekerjaan berhasil diupdate",
3   "success": true
4 }
```

g. Delete Pekerjaan (Admin)

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** http://localhost:3000/api/pekerjaan/11
- Body:**

```
{
  "_id": 13,
}
```
- Response:**

```
{
  "message": "Riwayat pekerjaan berhasil dihapus (soft delete)"
}
```

7. Link Github

<https://github.com/cococimoryy1/alumniproject-golang>