Nama: Daffa Surya Arrayan

Nim: 434231042

kelas: C5 - Pemrograman Backend lanjutan praktikum

Ambil Semua Data Alumni (GET)

```
func GetUsersAlumni(c *fiber.Ctx) error {
    rows, err := database.DB.Query("SELECT id, nim, nama, jurusan, angkatan, tahun_lulus, email, no_telepon, created_at, updated_at FROM al
    if err != nil {
        return c.Status(500).JSON(fiber.Map{"error": err.Error()})
    }
    defer rows.Close()

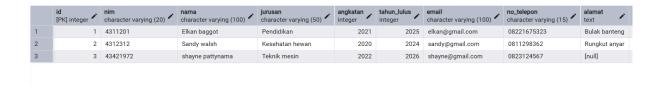
users := []models.Alumni{}
    for rows.Next() {
        var Alumni models.Alumni
        if err := rows.Scan(&Alumni.ID, &Alumni.NIM, &Alumni.Nama, &Alumni.Jurusan, &Alumni.Angkatan, &Alumni.Tahun_lulus, &Alumni.Email, &Auror c.Status(500).JSON(fiber.Map{"error": err.Error()})
    }
    users = append(users, Alumni)
}

return c.JSON(users)
}
```

### **Output pada Test API:**

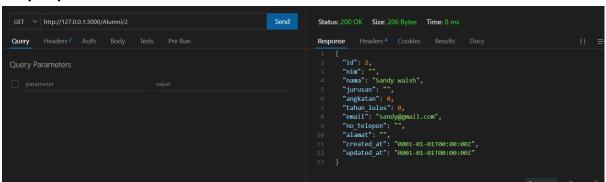
```
[
    "id": 3,
    "nim": "43421972",
    "nama": "shayne pattynama",
    "jurusan": "Teknik mesin",
    "angkatan": 2022,
    "tahun lulus": 2026,
    "email": "shayne@gmail.com",
    "no_telepon": "0823124567",
    "alamat": "",
    "created_at": "2025-09-06T16:28:01.737842",
    "updated_at": "2025-09-06T16:28:01.737842"
},
{
    "id": 1,
        "nim": "4311201",
        "nama": "Elkan baggot",
        "jurusan": "Pendidikan",
        "angkatan": 2021,
        "tahun_lulus": 2025,
        "email": "elkan@gmail.com",
        "no_telepon": "08221675323",
        "alamat": "",
        "created_at": "2025-09-06T15:52:15.181393Z",
        "updated_at": "2025-09-06T15:52:15.181393Z",
        "nim": "4312312",
        "nama": "83a104 walsh",
        "anama": "Sandy walsh",
        """)
```

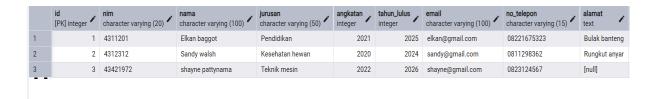
## Data Pada database Postgree:



# Ambil Data Alumni berdasarkan ID (GET)

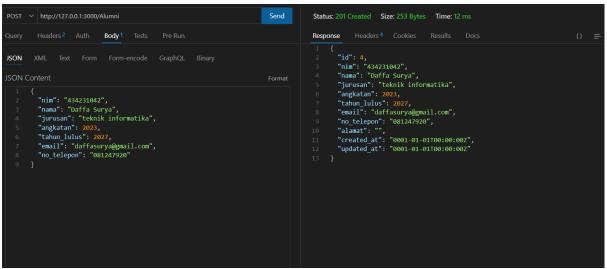
#### **Output pada Test API:**

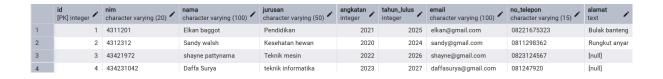




# Tambah Alumni baru ( POST )

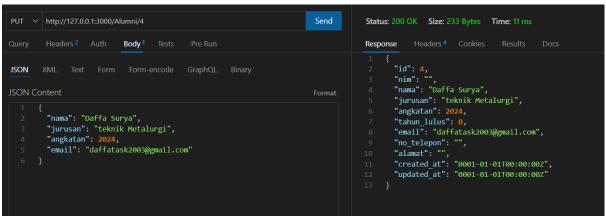
# **Output pada Test API:**

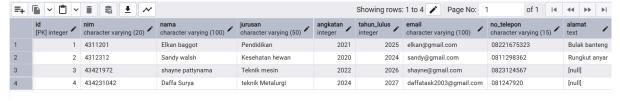




Update data alumni menggunakan ID ( PUT )

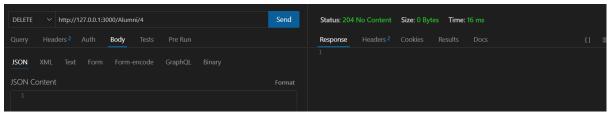
## Output pada Test API:

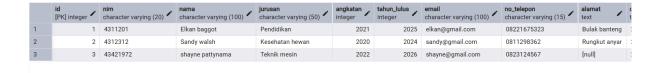




• Delete data alumni menggunakan ID ( Delete )

# **Output pada Test API:**





Ambil Semua Data Pekerjaan\_alumni (GET)

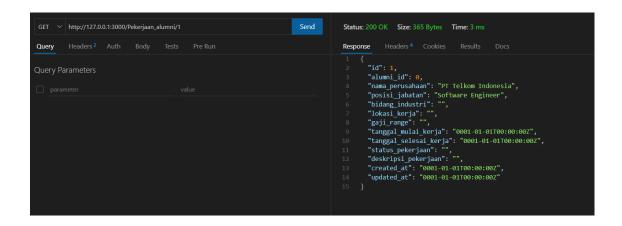
```
1 func CreatePekerjaanAlumni(c *fiber.Ctx) error {
       var pekerjaan models.Pekerjaan_Alm
       if err := c.BodyParser(&pekerjaan); err != nil {
           return c.Status(400).JSON(fiber.Map{"error": "Invalid request"})
         INSERT INTO pekerjaan_alumni (
              alumni_id,
               nama_perusahaan,
              posisi_jabatan,
              bidang_industri,
              lokasi_kerja,
               gaji_range,
              tanggal_mulai_kerja,
              tanggal_selesai_kerja,
               status_pekerjaan,
              deskripsi_pekerjaan
           ) VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10)
           RETURNING id
      err := database.DB.QueryRow(
         pekerjaan.Nama_Perusahaan,
         pekerjaan.Posisi_jabatan,
pekerjaan.Bidang_industri,
         pekerjaan.Lokasi_kerja,
         pekerjaan.Tanggal_kerja,
pekerjaan.Tanggal_selesai,
       ).Scan(pekerjaan.ID)
       if err != nil {
           return c.Status(500).JSON(fiber.Map{"error": err.Error()})
       return c.Status(201).JSON(fiber.Map{
           "message": "Data pekerjaan alumni berhasil ditambahkan",
                      pekerjaan.ID,
```

#### **Data Pada database Postgree:**



# Ambil Semua Data Pekerjaan\_alumni berdasarkan ID (GET)

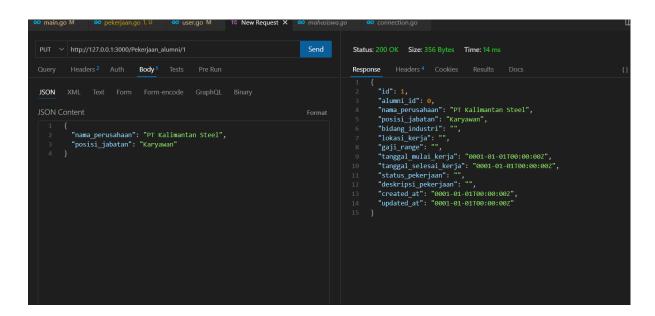
```
func GetPekerjaan_alumni_ID(c *fiber.Ctx) error {
   id := c.Params("id")
   var pekerjaan models.Pekerjaan_Alm
   err := database.DB.QueryRow("SELECT id, nama_perusahaan, posisi_jabatan FROM pekerjaan_alumni WHERE id = $1", id).Scan(&pekerjaan.ID, &pek
   if err != nil {
        return c.Status(404).JSON(fiber.Map{"error": "data tidak ditemukan"})
   }
   return c.JSON(pekerjaan)
}
```



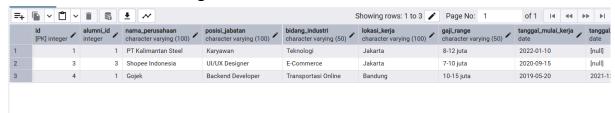
### Data Pada database Postgree:



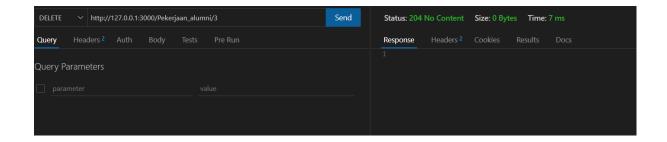
# Update data alumni menggunakan ID ( PUT )



#### **Data Pada database Postgree:**



# Delete Pekerjaan alumni Menggunakan ID ( Delete )



# Data Pada database Postgree:



# Tambah Pekerjaan\_alumni ( POST )

```
Send
                                                                                         Status: 201 Created Size: 400 Bytes Time: 23 ms
POST V http://127.0.0.1:3000/Pekerjaan_alumni
                                        Body 1
JSON
                                                                                                   "alumni_id": 3,
                                                                                                   "nama_perusahaan": "PT Maju Jaya",
                                                                                                   "posisi_jabatan": "Software Engineer",
"bidang_industri": "Teknologi Informasi",
            "alumni_id": 3,
                                                                                                   "lokasi_kerja": "Jakarta",
"gaji_range": "8 - 12 Juta",
"tanggal_mulai_kerja": "0001-01-01T00:00:00Z",
            "nama_perusahaan": "PT Maju Jaya",
"posisi_jabatan": "Software Engineer",
"bidang_industri": "Teknologi Informasi",
                                                                                                   "tanggal_selesai_kerja": "0001-01-01T00:00:00Z",
            "lokasi_kerja": "Jakarta",
"gaji_range": "8 - 12 Juta",
"status_pekerjaan": "aktif"
                                                                                               "status_pekerjaan": "aktif",

"deskripsi_pekerjaan": "",
                                                                                                   "created_at": "0001-01-01T00:00:00Z",
                                                                                                   "updated at": "0001-01-01T00:00:00Z
```



# Get Pekerjaan alumni berdasarkan alumni\_id (GET)

### **Output pada Test API:**