

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 alumni")
    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, &Alumni.No_telepon, &Alumni.Created_at, &Alumni.Updated_at); err != nil {
            return 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": "2025-09-06T16:28:01.73784Z",
    "updated_at": "2025-09-06T16:28:01.73784Z"
  },
  {
    "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"
  },
  {
    "id": 2,
    "nim": "4312312",
    "nama": "Sandy walsh",
    "jurusan": "Teknik komputer",
    "angkatan": 2022,
    "tahun_lulus": 2026,
    "email": "sandy@gmail.com",
    "no_telepon": "08221675323",
    "alamat": "",
    "created_at": "2025-09-06T15:52:15.181393Z",
    "updated_at": "2025-09-06T15:52:15.181393Z"
  }
]
```

Data Pada database Postgree:

| | id [PK] integer | nim character varying (20) | nama character varying (100) | jurusan character varying (50) | angkatan integer | tahun_lulus integer | email character varying (100) | no_telepon character varying (15) | alamat text |
|---|--------------------|-------------------------------|---------------------------------|-----------------------------------|---------------------|------------------------|----------------------------------|--------------------------------------|----------------|
| 1 | 1 | 4311201 | Elkan baggot | Pendidikan | 2021 | 2025 | elkan@gmail.com | 08221675323 | Bulak banteng |
| 2 | 2 | 4312312 | Sandy walsh | Kesehatan hewan | 2020 | 2024 | sandy@gmail.com | 0811298362 | Rungkut anyar |
| 3 | 3 | 43421972 | shayne pattynama | Teknik mesin | 2022 | 2026 | shayne@gmail.com | 0823124567 | [null] |

- Ambil Data Alumni berdasarkan ID (GET)

```
func GetUsersAlumniID(c *fiber.Ctx) error {
    id := c.Params("id")
    var Alumni models.Alumni
    err := database.DB.QueryRow("SELECT id, nama, email FROM alumni WHERE id = $1", id).Scan(&Alumni.ID, &Alumni.Nama, &Alumni.Email)
    if err != nil {
        return c.Status(404).JSON(fiber.Map{"error": "data tidak ditemukan"})
    }
    return c.JSON(Alumni)
}
```

Output pada Test API:

GET http://127.0.0.1:3000/Alumni/2 Send

Query Headers Auth Body Tests Pre Run

Query Parameters

parameter

value

Status: 200 OK Size: 206 Bytes Time: 8 ms

Response Headers Cookies Results Docs

```
1 {
2   "id": 2,
3   "nim": "",
4   "nama": "Sandy walsh",
5   "jurusan": "",
6   "angkatan": 0,
7   "tahun_lulus": 0,
8   "email": "sandy@gmail.com",
9   "no_telepon": "",
10  "alamat": "",
11  "created_at": "0001-01-01T00:00:00Z",
12  "updated_at": "0001-01-01T00:00:00Z"
13 }
```

Data Pada database Postgree:

| | id [PK] integer | nim character varying (20) | nama character varying (100) | jurusan character varying (50) | angkatan integer | tahun_lulus integer | email character varying (100) | no_telepon character varying (15) | alamat text |
|---|--------------------|-------------------------------|---------------------------------|-----------------------------------|---------------------|------------------------|----------------------------------|--------------------------------------|----------------|
| 1 | 1 | 4311201 | Elkan baggot | Pendidikan | 2021 | 2025 | elkan@gmail.com | 08221675323 | Bulak banteng |
| 2 | 2 | 4312312 | Sandy walsh | Kesehatan hewan | 2020 | 2024 | sandy@gmail.com | 0811298362 | Rungkut anyar |
| 3 | 3 | 43421972 | shayne pattynama | Teknik mesin | 2022 | 2026 | shayne@gmail.com | 0823124567 | [null] |

- **Tambah Alumni baru (POST)**

```
func CreateUserAlumni(c *fiber.Ctx) error {
    var Alumni models.Alumni
    if err := c.BodyParser(&Alumni); err != nil {
        return c.Status(400).JSON(fiber.Map{"error": "Invalid request"})
    }

    query := "INSERT INTO alumni (nim, nama, jurusan, angkatan, tahun_lulus, email, no_telepon, created_at, updated_at) VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9)"
    err := database.DB.QueryRow(query, Alumni.NIM, Alumni>Nama, Alumni.Jurusan, Alumni.Angkatan, Alumni.Tahun_lulus, Alumni.Email, Alumni.No_telepon, time.Now().Format("2006-01-02 15:04:05"), time.Now().Format("2006-01-02 15:04:05"))
    if err != nil {
        return c.Status(500).JSON(fiber.Map{"error": err.Error()})
    }

    return c.Status(201).JSON(Alumni)
}
```

Output pada Test API:

POST http://127.0.0.1:3000/Alumni Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "nim": "434231042",
3   "nama": "Daffa Surya",
4   "jurusan": "teknik informatika",
5   "angkatan": 2023,
6   "tahun_lulus": 2027,
7   "email": "daffasurya@gmail.com",
8   "no_telepon": "081247920"
9 }
```

Status: 201 Created Size: 253 Bytes Time: 12 ms

Response Headers 4 Cookies Results Docs

```
1 {
2   "id": 4,
3   "nim": "434231042",
4   "nama": "Daffa Surya",
5   "jurusan": "teknik informatika",
6   "angkatan": 2023,
7   "tahun_lulus": 2027,
8   "email": "daffasurya@gmail.com",
9   "no_telepon": "081247920",
10  "alamat": "",
11  "created_at": "0001-01-01T00:00:00Z",
12  "updated_at": "0001-01-01T00:00:00Z"
13 }
```

Data Pada database Postgree:

| | id [PK] Integer | nim character varying (20) | nama character varying (100) | jurusan character varying (50) | angkatan Integer | tahun_lulus Integer | email character varying (100) | no_telepon character varying (15) | alamat text |
|---|--------------------|-------------------------------|---------------------------------|-----------------------------------|---------------------|------------------------|----------------------------------|--------------------------------------|----------------|
| 1 | 1 | 4311201 | Elkan baggot | Pendidikan | 2021 | 2025 | elkan@gmail.com | 08221675323 | Bulak banteng |
| 2 | 2 | 4312312 | Sandy walsh | Kesehatan hewan | 2020 | 2024 | sandy@gmail.com | 0811298362 | Rungkut anyar |
| 3 | 3 | 43421972 | shayne pattynama | Teknik mesin | 2022 | 2026 | shayne@gmail.com | 0823124567 | [null] |
| 4 | 4 | 434231042 | Daffa Surya | teknik Informatika | 2023 | 2027 | daffasurya@gmail.com | 081247920 | [null] |

- Update data alumni menggunakan ID (PUT)

```
func UpdateUserAlumni(c *fiber.Ctx) error {
    id := c.Params("id")
    var Alumni models.Alumni
    if err := c.BodyParser(&Alumni); err != nil {
        return c.Status(400).JSON(fiber.Map{"error": "Invalid request"})
    }

    query := "UPDATE alumni SET nama = $1, jurusan = $2, angkatan = $3, email = $4, updated_at = $5 WHERE id = $6"
    _, err := database.DB.Exec(query, Alumni>Nama, Alumni.Jurusan, Alumni.Angkatan, Alumni.Email, time.Now(), id)
    if err != nil {
        return c.Status(500).JSON(fiber.Map{
            "error": "Gagal menghapus data alumni",
        })
    }

    Alumni.ID = atoi(id)
    return c.JSON(Alumni)
}
```

Output pada Test API:

PUT

http://127.0.0.1:3000/Alumni/4

Send

Status: 200 OK Size: 233 Bytes Time: 11 ms

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content

Format

```
1 {
2   "nama": "Daffa Surya",
3   "jurusan": "teknik Metalurgi",
4   "angkatan": 2024,
5   "email": "daffatask2003@gmail.com"
6 }
```

Response Headers 4 Cookies Results Docs

```
1 {
2   "id": 4,
3   "nim": "",
4   "nama": "Daffa Surya",
5   "jurusan": "teknik Metalurgi",
6   "angkatan": 2024,
7   "tahun_lulus": 0,
8   "email": "daffatask2003@gmail.com",
9   "no_telepon": "",
10  "alamat": "",
11  "created_at": "0001-01-01T00:00:00Z",
12  "updated_at": "0001-01-01T00:00:00Z"
13 }
```

Data Pada database Postgree:

| Showing rows: 1 to 4 Page No: 1 of 1 | | | | | | | | | |
|--------------------------------------|--------------------|-------------------------------|---------------------------------|-----------------------------------|---------------------|------------------------|----------------------------------|--------------------------------------|----------------|
| | id [PK] integer | nim character varying (20) | nama character varying (100) | jurusan character varying (50) | angkatan integer | tahun_lulus integer | email character varying (100) | no_telepon character varying (15) | alamat text |
| 1 | 1 | 4311201 | Elkan baggot | Pendidikan | 2021 | 2025 | elkan@gmail.com | 08221675323 | Bulak banteng |
| 2 | 2 | 4312312 | Sandy walsh | Kesehatan hewan | 2020 | 2024 | sandy@gmail.com | 0811298362 | Rungkut anyar |
| 3 | 3 | 43421972 | shayne pattynama | Teknik mesin | 2022 | 2026 | shayne@gmail.com | 0823124567 | [null] |
| 4 | 4 | 434231042 | Daffa Surya | teknik Metalurgi | 2024 | 2027 | daffatask2003@gmail.com | 081247920 | [null] |

- Delete data alumni menggunakan ID (Delete)

```
func DeleteUserAlumni(c *fiber.Ctx) error {
    id := c.Params("id")
    _, err := database.DB.Exec("DELETE FROM alumni WHERE id = $1", id)
    if err != nil {
        // return c.Status(500).JSON(fiber.Map{"error": err.Error()})
        return c.Status(500).JSON(fiber.Map{
            "error": "Gagal menghapus data alumni",
        })
    }
    return c.SendStatus(204)
}
```

Output pada Test API:

The screenshot shows a REST client interface. The top bar indicates a DELETE request to `http://127.0.0.1:3000/Alumni/4` with a status of `204 No Content`, size of `0 Bytes`, and time of `16 ms`. The 'Body' tab is selected, showing a 'JSON Content' section with a 'Format' button. The response is displayed in the 'Response' tab, showing a single empty line.

Data Pada database Postgree:

| | id [PK] integer | nim character varying (20) | nama character varying (100) | jurusan character varying (50) | angkatan integer | tahun_lulus integer | email character varying (100) | no_telepon character varying (15) | alamat text |
|---|--------------------|-------------------------------|---------------------------------|-----------------------------------|---------------------|------------------------|----------------------------------|--------------------------------------|----------------|
| 1 | 1 | 4311201 | Elkan baggot | Pendidikan | 2021 | 2025 | elkan@gmail.com | 08221675323 | Bulak banteng |
| 2 | 2 | 4312312 | Sandy walsh | Kesehatan hewan | 2020 | 2024 | sandy@gmail.com | 0811298362 | Rungkut anyar |
| 3 | 3 | 43421972 | shayne pattynama | Teknik mesin | 2022 | 2026 | shayne@gmail.com | 0823124567 | [null] |

- Ambil Semua Data Pekerjaan_alumni (GET)

```
1 func CreatePekerjaanAlumni(c *fiber.Ctx) error {
2     var pekerjaan models.Pekerjaan_Alm
3
4     if err := c.BodyParser(&pekerjaan); err != nil {
5         return c.Status(400).JSON(fiber.Map{"error": "Invalid request"})
6     }
7
8     query := `
9         INSERT INTO pekerjaan_alumni (
10             alumni_id,
11             nama_perusahaan,
12             posisi_jabatan,
13             bidang_industri,
14             lokasi_kerja,
15             gaji_range,
16             tanggal_mulai_kerja,
17             tanggal_selesai_kerja,
18             status_pekerjaan,
19             deskripsi_pekerjaan
20         ) VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10)
21         RETURNING id
22     `
23
24     err := database.DB.QueryRow(
25         query,
26         pekerjaan.Alumni_ID,
27         pekerjaan>Nama_Perusahaan,
28         pekerjaan.Posisi_jabatan,
29         pekerjaan.Bidang_industri,
30         pekerjaan.Lokasi_kerja,
31         pekerjaan.Gaji_range,
32         pekerjaan.Tanggal_kerja,
33         pekerjaan.Tanggal_selesai,
34         pekerjaan.Status,
35         pekerjaan.Deskripsi,
36     ).Scan(pekerjaan.ID)
37
38     if err != nil {
39         return c.Status(500).JSON(fiber.Map{"error": err.Error()})
40     }
41
42     return c.Status(201).JSON(fiber.Map{
43         "message": "Data pekerjaan alumni berhasil ditambahkan",
44         "id":      pekerjaan.ID,
45     })
46 }
```

Output pada Test API:

GET http://127.0.0.1:3000/Pekerjaan_alumni

Send

Query

Headers 2AuthBodyTestsPre Run

Query Parameters

parameter

value

Status: 200 OKSize: 1.38 KBTime: 3 ms

Response

Headers 4CookiesResultsDocs

1

[

2

{

3

"id": 1,

4

"alumni_id": 1,

5

"nama_perusahaan": "PT Telkom Indonesia",

6

"posisi_jabatan": "Software Engineer",

7

"bidang_industri": "Teknologi",

8

"lokasi_kerja": "Jakarta",

9

"gaji_range": "8-12 juta",

10

"tanggal_mulai_kerja": "2022-01-10T00:00:00Z",

11

"tanggal_selesai_kerja": "0001-01-01T00:00:00Z",

12

"status_pekerjaan": "aktif",

13

"deskripsi_pekerjaan": "Mengembangkan aplikasi berbasis web menggunakan Go dan PostgreSQL",

14

"created_at": "2025-09-06T20:19:35.434595Z",

15

"updated_at": "2025-09-06T20:19:35.434595Z"

16

},

17

{

18

"id": 3,

19

"alumni_id": 3,

20

"nama_perusahaan": "Shopee Indonesia",

21

"posisi_jabatan": "UI/UX Designer",

22

"bidang_industri": "E-Commerce",

23

"lokasi_kerja": "Jakarta",

24

"gaji_range": "7-10 juta",

25

"tanggal_mulai_kerja": "2020-09-15T00:00:00Z",

26

"tanggal_selesai_kerja": "0001-01-01T00:00:00Z",

27

"status_pekerjaan": "aktif",

28

"deskripsi_pekerjaan": "Merancang tampilan antarmuka aplikasi mobile dan website",

29

"created_at": "2025-09-06T20:19:35.434595Z",

Data Pada database Postgree:

| Showing rows: 1 to 3Page No: 1 of 1 | | | | | | | | | | |
|-------------------------------------|--------------------|----------------------|--|---|---|---|--------------------------------------|-----------------------------|--------------|--------|
| | id [PK] integer | alumni_id integer | nama_perusahaan character varying (100) | posisi_jabatan character varying (100) | bidang_industri character varying (50) | lokasi_kerja character varying (100) | gaji_range character varying (50) | tanggal_mulai_kerja date | tang date | |
| 1 | 1 | 1 | PT Telkom Indonesia | Software Engineer | Teknologi | Jakarta | 8-12 juta | 2022-01-10 | | [null] |
| 2 | 2 | 3 | Shopee Indonesia | UI/UX Designer | E-Commerce | Jakarta | 7-10 juta | 2020-09-15 | | [null] |
| 3 | 3 | 4 | Gojek | Backend Developer | Transportasi Online | Bandung | 10-15 juta | 2019-05-20 | 202 | |

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)
}
```

Output pada Test API:

GET http://127.0.0.1:3000/Pekerjaan_alumni/1

Send

Query

Headers 2 Auth Body Tests Pre Run

Query Parameters

☐ parameter

value

Status: 200 OK Size: 365 Bytes Time: 3 ms

Response

Headers 4 Cookies Results Docs

```
1 {
2   "id": 1,
3   "alumni_id": 0,
4   "nama_perusahaan": "PT Telkom Indonesia",
5   "posisi_jabatan": "Software Engineer",
6   "bidang_industri": "",
7   "lokasi_kerja": "",
8   "gaji_range": "",
9   "tanggal_mulai_kerja": "0001-01-01T00:00:00Z",
10  "tanggal_selesai_kerja": "0001-01-01T00:00:00Z",
11  "status_pekerjaan": "",
12  "deskripsi_pekerjaan": "",
13  "created_at": "0001-01-01T00:00:00Z",
14  "updated_at": "0001-01-01T00:00:00Z"
15 }
```

Data Pada database Postgree:

| | id [PK] integer | alumni_id integer | nama_perusahaan character varying (100) | posisi_jabatan character varying (100) | bidang_industri character varying (50) | lokasi_kerja character varying (100) | gaji_range character varying (50) | tanggal_mulai_kerja date | tangg date |
|---|--------------------|----------------------|--|---|---|---|--------------------------------------|-----------------------------|---------------|
| 1 | | 1 | PT Telkom Indonesia | Software Engineer | Teknologi | Jakarta | 8-12 juta | 2022-01-10 | [null] |
| 2 | | 3 | Shopee Indonesia | UI/UX Designer | E-Commerce | Jakarta | 7-10 juta | 2020-09-15 | [null] |
| 3 | | 4 | Gojek | Backend Developer | Transportasi Online | Bandung | 10-15 juta | 2019-05-20 | 2021- |

Update data alumni menggunakan ID (PUT)

```
func UpdatePekerjaanAlumni(c *fiber.Ctx) error {
    id := c.Params("id")
    var Job models.Pekerjaan_Alm
    if err := c.BodyParser(&Job); err != nil {
        return c.Status(400).JSON(fiber.Map{"error": "Invalid request"})
    }

    // query := "UPDATE alumni SET nama = $1, jurusan = $2, angkatan = $3, email = $4, updated_at = $5 WHERE id = $6"
    query := "UPDATE pekerjaan_alumni SET nama_perusahaan = $1, posisi_jabatan = $2, updated_at = $3 WHERE id = $4"
    _, err := database.DB.Exec(query, Job>Nama_Perusahaan, Job.Posisi_jabatan, time.Now(), id)

    if err != nil {
        return c.Status(500).JSON(fiber.Map{
            "error": "Gagal menghapus data alumni",
        })
    }

    Job.ID = atoi(id)
    return c.JSON(Job)
}
```


Output pada Test API:

The screenshot shows a REST client interface with a PUT request to `http://127.0.0.1:3000/Pekerjaan_alumni/1`. The request body is a JSON object: `{ "nama_perusahaan": "PT Kalimantan Steel", "posisi_jabatan": "Karyawan" }`. The response status is `200 OK` with a size of `356 Bytes` and a time of `14 ms`. The response body is a JSON object: `{ "id": 1, "alumni_id": 0, "nama_perusahaan": "PT Kalimantan Steel", "posisi_jabatan": "Karyawan", "bidang_industri": "", "lokasi_kerja": "", "gaji_range": "", "tanggal_mulai_kerja": "0001-01-01T00:00:00Z", "tanggal_selesai_kerja": "0001-01-01T00:00:00Z", "status_pekerjaan": "", "deskripsi_pekerjaan": "", "created_at": "0001-01-01T00:00:00Z", "updated_at": "0001-01-01T00:00:00Z" }`.

Data Pada database Postgree:

| Showing rows: 1 to 3 | | | | | | | | | |
|----------------------|--------------------|----------------------|--|---|---|---|--------------------------------------|-----------------------------|-----------------|
| | id [PK] integer | alumni_id integer | nama_perusahaan character varying (100) | posisi_jabatan character varying (100) | bidang_industri character varying (50) | lokasi_kerja character varying (100) | gaji_range character varying (50) | tanggal_mulai_kerja date | tanggal date |
| 1 | 1 | 1 | PT Kalimantan Steel | Karyawan | Teknologi | Jakarta | 8-12 juta | 2022-01-10 | [null] |
| 2 | 3 | 3 | Shopee Indonesia | UI/UX Designer | E-Commerce | Jakarta | 7-10 juta | 2020-09-15 | [null] |
| 3 | 4 | 1 | Gojek | Backend Developer | Transportasi Online | Bandung | 10-15 juta | 2019-05-20 | 2021-10-20 |

Delete Pekerjaan alumni Menggunakan ID (Delete)

```
func DeletePekerjaan_alumni(c *fiber.Ctx) error {
    id := c.Params("id")
    _, err := database.DB.Exec("DELETE FROM pekerjaan_alumni WHERE id = $1", id)
    if err != nil {
        // return c.Status(500).JSON(fiber.Map{"error": err.Error()})
        return c.Status(500).JSON(fiber.Map{
            "error": "Gagal menghapus data Pekerjaan alumni",
        })
    }
    return c.SendStatus(204)
}
```

Output pada Test API:

DELETE

http://127.0.0.1:3000/Pekerjaan_alumni/3

Send

Status: 204 No Content

Size: 0 Bytes

Time: 7 ms

Query

Headers 2

Auth

Body

Tests

Pre Run

Response

Headers 2

Cookies

Results

Docs

Query Parameters

☐

parameter

value

1

Data Pada database Postgree:

| | id [PK] integer | alumni_id integer | nama_perusahaan character varying (100) | posisi_jabatan character varying (100) | bidang_industri character varying (50) | lokasi_kerja character varying (100) | gaji_range character varying (50) | tanggal_mulai_kerja date | tanggal_selesai_kerja date |
|---|--------------------|----------------------|--|---|---|---|--------------------------------------|-----------------------------|-------------------------------|
| 1 | 1 | 1 | PT Kalimantan Steel | Karyawan | Teknologi | Jakarta | 8-12 juta | 2022-01-10 | [null] |
| 2 | 4 | 1 | Gojek | Backend Developer | Transportasi Online | Bandung | 10-15 juta | 2019-05-20 | 2021-12-31 |

Tambah Pekerjaan_alumni (POST)

```
1 func CreatePekerjaanAlumni(c *fiber.Ctx) error {
2     var job models.Pekerjaan_Alm
3     if err := c.BodyParser(&job); err != nil {
4         return c.Status(400).JSON(fiber.Map{
5             "error": "Invalid request body",
6         })
7     }
8
9     // Query INSERT
10    query := `
11    INSERT INTO pekerjaan_alumni
12    (alumni_id, nama_perusahaan, posisi_jabatan, bidang_industri, lokasi_kerja, gaji_range, tanggal_mulai_kerja, tanggal_selesai_kerja, status_pekerjaan, created_at, updated_at)
13    VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10, $11)
14    RETURNING id
15    `
16
17    err := database.DB.QueryRow(
18        query,
19        job.Alumni_ID,
20        job>Nama_Perusahaan,
21        job.Posisi_Jabatan,
22        job.Bidang_Industri,
23        job.Lokasi_Kerja,
24        job.Gaji_Range,
25        job.Tanggal_Kerja,
26        job.Tanggal_Seleasai,
27        job.Status,
28        time.Now(),
29        time.Now(),
30    ).Scan(&job.ID)
31
32    if err != nil {
33        return c.Status(500).JSON(fiber.Map{
34            "error": err.Error(),
35        })
36    }
37
38    return c.Status(201).JSON(job)
39 }
```

Output pada Test API:

POST

http://127.0.0.1:3000/Pekerjaan_alumni

Send

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Bin

JSON Content

Format

```
1 {
2   "alumni_id": 3,
3   "nama_perusahaan": "PT Maju Jaya",
4   "posisi_jabatan": "Software Engineer",
5   "bidang_industri": "Teknologi Informasi",
6   "lokasi_kerja": "Jakarta",
7   "gaji_range": "8 - 12 Juta",
8   "status_pekerjaan": "aktif"
9 }
```

Status: 201 Created

Size: 400 Bytes

Time: 23 ms

Response

Headers 4

Cookies

Results

Docs

{}

≡

```
1 {
2   "id": 9,
3   "alumni_id": 3,
4   "nama_perusahaan": "PT Maju Jaya",
5   "posisi_jabatan": "Software Engineer",
6   "bidang_industri": "Teknologi Informasi",
7   "lokasi_kerja": "Jakarta",
8   "gaji_range": "8 - 12 Juta",
9   "tanggal_mulai_kerja": "0001-01-01T00:00:00Z",
10  "tanggal_selesai_kerja": "0001-01-01T00:00:00Z",
11  "status_pekerjaan": "aktif",
12  "deskripsi_pekerjaan": "",
13  "created_at": "0001-01-01T00:00:00Z",
14  "updated_at": "0001-01-01T00:00:00Z"
15 }
```

Data Pada database Postgree:

| | id [PK] integer | alumni_id integer | nama_perusahaan character varying (100) | posisi_jabatan character varying (100) | bidang_industri character varying (50) | lokasi_kerja character varying (100) | gaji_range character varying (50) | tanggal_mulai_kerja date | tangg date |
|---|--------------------|----------------------|--|---|---|---|--------------------------------------|-----------------------------|---------------|
| 1 | 1 | 1 | PT Kalimantan Steel | Karyawan | Teknologi | Jakarta | 8-12 juta | 2022-01-10 | [null] |
| 2 | 4 | 1 | Gojek | Backend Developer | Transportasi Online | Bandung | 10-15 juta | 2019-05-20 | 2021- |
| 3 | 9 | 3 | PT Maju Jaya | Software Engineer | Teknologi Informasi | Jakarta | 8 - 12 Juta | 0001-01-01 | 0001- |

Get Pekerjaan alumni berdasarkan alumni_id (GET)

```
1 func GetPekerjaanByAlumni(c *fiber.Ctx) error {
2     alumniID := c.Params("alumni_id")
3
4     rows, err := database.DB.Query(`
5         SELECT id, nama_perusahaan, posisi_jabatan, bidang_industri, lokasi_kerja,
6             gaji_range, tanggal_mulai_kerja, tanggal_selesai_kerja, status_pekerjaan,
7             created_at, updated_at
8         FROM pekerjaan_alumni
9         WHERE alumni_id = $1
10    `, alumniID)
11
12    if err != nil {
13        return c.Status(500).JSON(fiber.Map{"error": err.Error()})
14    }
15    defer rows.Close()
16
17    var pekerjaanList []models.Pekerjaan_Alm
18
19    for rows.Next() {
20        var pekerjaan models.Pekerjaan_Alm
21        if err := rows.Scan(
22            &pekerjaan.ID,
23            &pekerjaan>Nama_Perusahaan,
24            &pekerjaan.Posisi_Jabatan,
25            &pekerjaan.Bidang_Industri,
26            &pekerjaan.Lokasi_Kerja,
27            &pekerjaan.Gaji_Range,
28            &pekerjaan.Tanggal_Kerja,
29            &pekerjaan.Tanggal_Selesai,
30            &pekerjaan.Status,
31            &pekerjaan.CreatedAt,
32            &pekerjaan.UpdatedAt,
33        ); err != nil {
34            return c.Status(500).JSON(fiber.Map{"error": err.Error()})
35        }
36        pekerjaanList = append(pekerjaanList, pekerjaan)
37    }
38
39    return c.JSON(pekerjaanList)
40 }
```

Output pada Test API:

| | | | | | |
|--|---|-------|----------------------|-----------------|------------|
| GET | http://127.0.0.1:3000/Pekerjaan_alumni/alumni/3 | Send | Status: 200 OK | Size: 416 Bytes | Time: 6 ms |
| Query | Headers ² | Auth | Body | Tests | Pre Run |
| Query Parameters | | | | | |
| <input type="checkbox"/> | parameter | value | | | |
| Response | | | Headers ⁴ | Cookies | Results |
| 1 [| | | | | |
| 2 { | | | | | |
| 3 "id": 0, | | | | | |
| 4 "alumni_id": 0, | | | | | |
| 5 "nama_perusahaan": "PT Maju Jaya", | | | | | |
| 6 "posisi_jabatan": "Software Engineer", | | | | | |
| 7 "bidang_industri": "Teknologi Informasi", | | | | | |
| 8 "lokasi_kerja": "Jakarta", | | | | | |
| 9 "gaji_range": "8 - 12 Juta", | | | | | |
| 10 "tanggal_mulai_kerja": "0001-01-01T00:00:00Z", | | | | | |
| 11 "tanggal_selesai_kerja": "0001-01-01T00:00:00Z", | | | | | |
| 12 "status_pekerjaan": "aktif", | | | | | |
| 13 "deskripsi_pekerjaan": "", | | | | | |
| 14 "created_at": "2025-09-07T11:19:18.555805Z", | | | | | |
| 15 "updated_at": "2025-09-07T11:19:18.555805Z" | | | | | |
| 16 } | | | | | |
| 17] | | | | | |

