

ExpressJs Routing Query dan Params

Sio Jurnalís Pipin, S.Kom.



Routing: Query Parameter

- Ketika melakukan permintaan (request) terkadang bisa jadi **meminta**, atau juga **meminta sekaligus mengirim value** ke server yaitu melalui url, atau form (html).
- Berfungsi untuk mengirim dan mengambil value yang dikirim melalui request
- Merupakan Query String yang bagian dari URL (**Uniform Resource Locator**) setelah tanda tanya (?) atau garis miring (/). Ini bertujuan untuk mengirim sejumlah kecil informasi ke server melalui url. Informasi ini biasanya digunakan sebagai parameter untuk mengkueri database, atau mungkin untuk memfilter hasil.

Request Query

- Request query dalam Expressjs mengidentifikasi query (key) pada url, contoh:

<http://localhost:3000/search?nama=budi&nim=19111>

- Implementasi pada ExpressJs

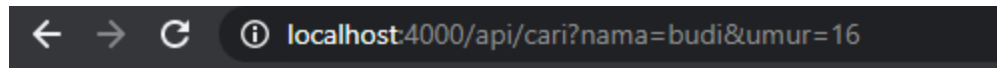
Pada index.js buat route:

```
21 app.get("/api/cari", function (req, res, next) {  
22   var nama = req.query.nama;  
23   console.log(`nama : ${nama}`);  
24   var umur = req.query.umur;  
25   console.log(`umur : ${umur}`);  
26   res.send(umur);  
27 });
```

```
SIO@DESKTOP-U2DMK91 MINGW64 ~/OneDrive - mikroskil.ac.id  
$ node index.js  
Server run  
nama : budi  
umur : 16  
GET /api/cari?nama=budi&umur=16 200 7.512 ms - 2
```

- Ini dapat di akses pada :

<http://localhost:4000/api/cari?nama=budi&umur=16>



Request Params

- Berfungsi untuk menangkap nilai yang dikirimkan melalui url yang mengirimkan nilai secara langsung tanpa key, contoh:

<http://localhost:3000/budi/19111>

- Implementasi pada ExpressJs

Pada index.js buat route:

```
//route dengan method get
app.get("/api/:nim/:nama", function (req, res) {
  res.statusCode = 200;
  //content-type pada expressjs
  res.setHeader("Content-Type", "text/plain");
  res.send(req.params);
});
```



- Ini dapat di akses pada :
- <http://localhost:4000/api/budi/16>