

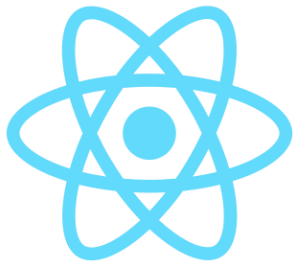
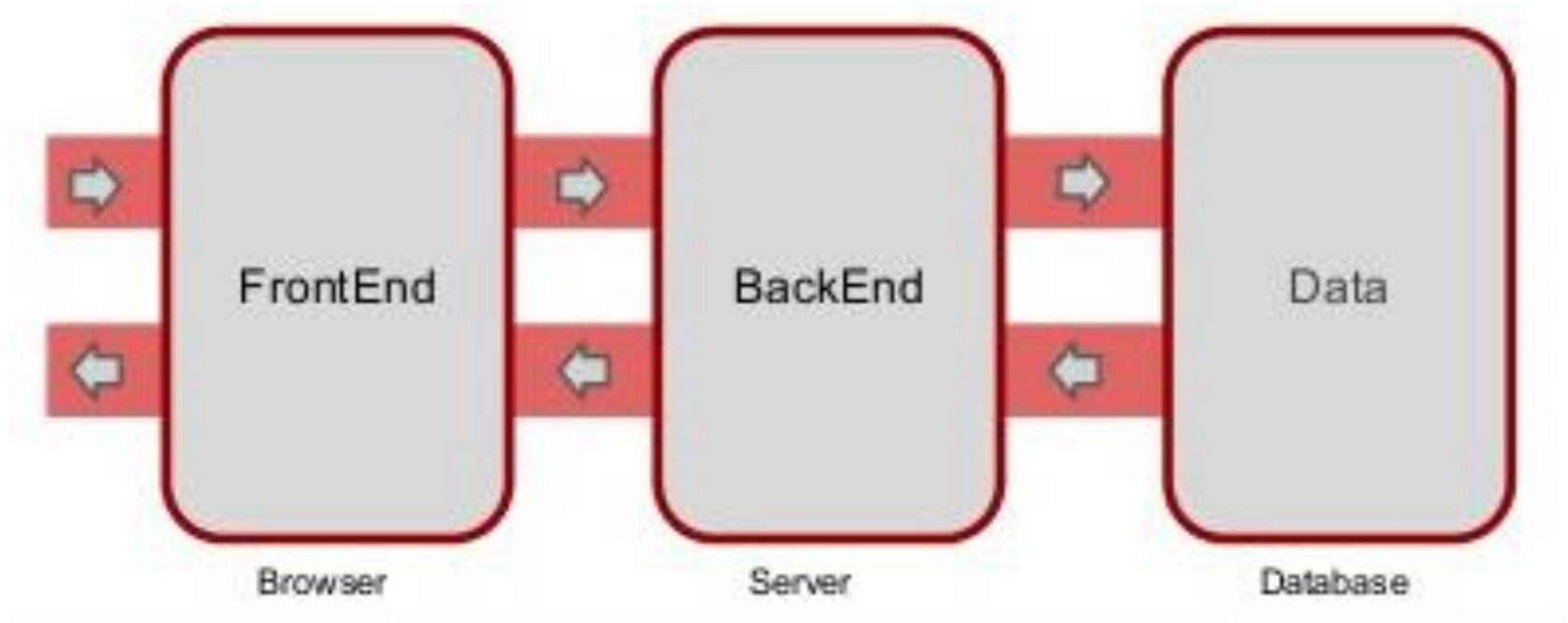
Back-End Development

Connecting

#12b Loopback & MongoDB



Web App Architecture



Loopback & MongoDB



mongoDB

#1 Activate MongoDB Server

■ Open terminal:

```
$ cd C:\Program Files\MongoDB\Server\3.6\bin
```

```
$ mongod -dbpath C:\data\db //jika belum//
```

```
$ mongod
```

■ If OK, it will show “Waiting for connections on port 27017”. Then open new terminal:

```
$ cd C:\Program Files\MongoDB\Server\3.6\bin
```

```
$ mongo
```

#2 Create Db, User & Collection

- Membuat db toko:

```
$ use toko
```

- Membuat db user:

```
$ db.createUser({user:"lintang",  
pwd:"1234", roles:["readWrite",  
"dbAdmin"]});
```

- Membuat collection “karyawan”:

```
$ db.createCollection('karyawan')
```

#3 Create Loopback App

- Create a Loopback application:

```
$ npm install -g loopback-cli
```

```
$ lb
```

```
$ cd lin_loopback
```

- Comment line *server.enableAuth();* on `/server/boot/authentication.js`

- Activate its server by run `$ node .`

```
Web server listening at:  
http://localhost:3000
```

```
Browse your REST API at  
http://localhost:3000/explorer
```

#4 Set MongoDB Connector

■ Install MongoDB connector:

```
$ npm install --save loopback-connector-mongodb
```

```
$ lb datasource
```

```
? Enter the datasource name: mongo //up to you!
```

```
? Select the connector for mongo: MongoDB (StrongLoop)
```

```
? Connection String url to override other settings:
```

```
mongodb://lintang:1234@localhost:27017/toko
```

```
? host: localhost
```

```
? port: 27017
```

```
? user: lintang
```

```
? password: 1234
```

```
? database: toko
```

#5 Create Loopback Model

■ On directory project, type:
\$ lb model

- [?] Enter model name: **karyawan //model = collection**
- [?] Select datasource to attach to: **mongo (mongoDB)**
- [?] Select model's base class (**PersistedModel**)
- [?] Expose person via the REST API? **Yes**
- [?] Custom plural form (used to build REST URL):
- [?] Common model or server only? **Common**

Let's add some person properties now.

Enter an empty property name when done.

- [?] Property name: **nama**
- [?] Property type: (Use arrow keys) **string**
- [?] Required? (y/N) **y**

#6 Try to POST & GET

- Run **\$ node .** then try to POST & GET directly on Loopback explorer (**karyawan model**) or using Postman, then see your MongoDB collections (**karyawan collection**)!

The screenshot displays the LoopBack API Explorer interface. The top bar is dark green with the text "LoopBack API Explorer" on the left, "Token Not Set" in the center, and a "Set Access Token" button on the right. Below this, the "Response Body" section is highlighted in light blue. It contains a JSON array of two employee objects. To the right of the JSON, a terminal window titled "C:\WINDOWS\system32\cmd.exe - mongo" shows the MongoDB command `db.karyawan.find().pretty()` and its output, which matches the JSON in the response body.

LoopBack API Explorer Token Not Set Set Access Token

Response Body

```
[
  {
    "nama": "Andi",
    "usia": 23,
    "id": "5aadd651b28fee1d80f5bed1"
  },
  {
    "nama": "Budi",
    "usia": 25,
    "id": "5aadd698b28fee1d80f5bed2"
  }
]
```

```
C:\WINDOWS\system32\cmd.exe - mongo
"nama" : "Andi",
"usia" : 23
}
> db.karyawan.find().pretty()
{
  "_id" : ObjectId("5aadd651b28fee1d80f5bed1"),
  "nama" : "Andi",
  "usia" : 23
}
{
  "_id" : ObjectId("5aadd698b28fee1d80f5bed2"),
  "nama" : "Budi",
  "usia" : 25
}
>
```

Loopback & mLab



mLab

Back-End Development

Connecting

#12b Loopback & MongoDB

