Back-End Development

Exploration

Purwadhika
Startup and Coding School

Database

- A **database** is *an organized collection of data*.
- The **main purpose** of database is to operate large amount of information by storing, retrieving and managing.
- There are many dynamic websites on the world wide web nowadays which are handled through databases. For example, a model to checks the availability of rooms in a hotel. It is an example of dynamic website that uses database.

Purwadhika
Startup and Coding School

Non-RDBMS / NoSQL

- **NoSQL** or **Non-RDBMS** (*Non Relational Database Management Systems*) is database that is not modeled like relational model based.
- The data structure used by NoSQL databases are different from relational databases. It's faster and more flexible than relational db. That's why it's increasingly used in *big data* & *real time web application*.
- Here are the example of NoSQL database: MongoDB, Cassandra, CouchDB, OrientDB, IBM Domino, ArangoDB & Apache Ignite.

MongoDB

MongoDB is free & open source cross-platform document-oriented database. It's classified as NoSQL database & uses JSON-like documents with schemas.

MongoDB has been developed by MongoDB Inc. since 11th February 2009, and is published under GNU Affero General Public License & the Apache License.

Purwadhika
Startup and Coding School

mongoDB

MongoDB Ranking

5th All DB-engines

341 systems in ranking, March 2018

Rank Mar 2018	Rank Feb 2018	Rank Mar 2017	DBMS	Database Model	Score Mar 2018	Score Feb 2018	Score Mar 2017
1.	1.	1.	Oracle	Relational DBMS	1289.61	-13.67	-109.89
2.	2.	2.	MySQL	Relational DBMS	1228.87	-23.60	-147.21
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1104.79	-17.25	-102.70
4.	4.	4.	PostgreSQL	Relational DBMS	399.35	+10.97	+41.71
5.	5.	5.	MongoDB	Document store	340.52	+4.10	+13.59
6.	6.	6.	DB2	Relational DBMS	186.66	-3.31	+1.75
7.	7.	7.	Microsoft Access	Relational DBMS	131.95	+1.88	-0.99
8.	8.	↑ 10.	Redis	Key-value store	131.22	+4.21	+18.22
9.	9.	↑ 11.	Elasticsearch	Search engine	128.54	+3.23	+22.32
10.	10.	↓ 8.	Cassandra	Wide column store	123.49	+0.71	-5.70

https://db-engines.com/en/ranking

Purwadhika
Startup and Coding School

mongoDB

MongoDB Ranking

1st Doc Store DB-engines

44 systems in ranking, March 2018

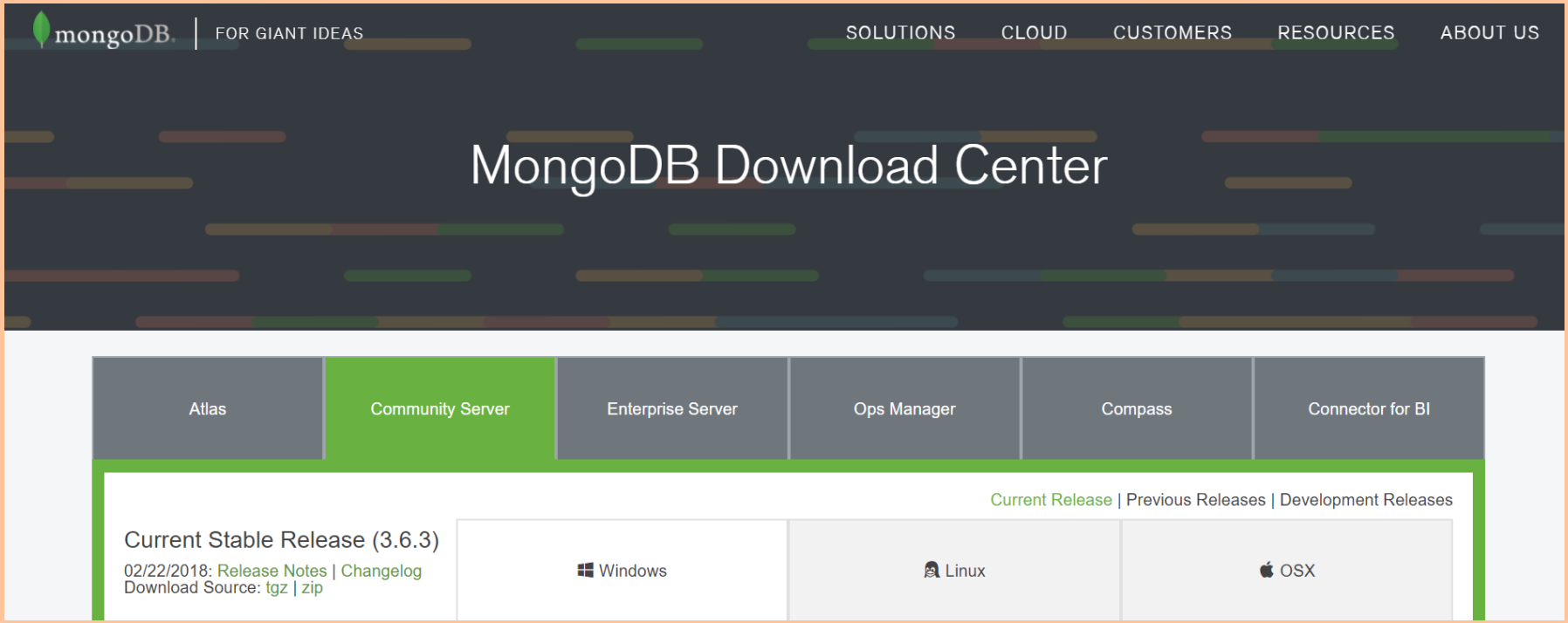
Rank Mar 2018	Rank Feb 2018	Rank Mar 2017	DBMS	Database Model	Score Mar 2018	Score Feb 2018	Score Mar 2017
1.	1.	1.	MongoDB	Document store	340.52	+4.10	+13.59
2.	2.	2.	Amazon DynamoDB	Multi-model	42.46	+2.58	+11.33
3.	3.	3.	Couchbase	Document store	32.90	+1.15	+2.86
4.	4.	4.	CouchDB	Document store	20.20	-0.09	-2.73
5.	5.	↑ 9.	Microsoft Azure Cosmos DB	Multi-model	16.76	+0.57	+12.81
6.	6.	↓ 5.	MarkLogic	Multi-model	10.97	-0.05	-0.13
7.	↑ 8.	↑ 13.	Firebase Realtime Database	Document store	6.59	+0.75	+3.80
8.	↓ 7.	↓ 7.	OrientDB	Multi-model	6.46	+0.52	+1.12
9.	9.	↓ 6.	RethinkDB	Document store	4.81	+0.12	-0.73
10.	10.	↓ 8.	Cloudant	Document store	4.30	+0.60	-0.69

https://db-engines.com/en/ranking

Purwadhika
Startup and Coding School

Getting Started

- Create ***data*** & ***db*** folder at ***C:/data/db***
- Download & install MongoDB



Activate Server

- Open terminal:

```
$ cd C:\Program Files\MongoDB\Server\3.6\bin
$ mongod –dbpath C:\data\db
$ 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
```

Purwadhika
Startup and Coding School

How to Work With MongoDB Database

Purwadhika
Startup and Coding School

mongoDB

Database

- Menampilkan daftar database:

```
$ show dbs
```

- Menampilkan db aktif (default: test):

```
$ db
```

- Membuat sekaligus menggunakan db “toko”:

```
$ use toko
```

- Hapus db “toko”:

```
$ use toko
$ db.dropDatabase()
```

Purwadhika
Startup and Coding School

mongoDB

Collection

- Membuat db user:

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

- Membuat collection “karyawan”:

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

- Menampilkan daftar collection dalam db:

```
$ show collections
```

- Hapus collection “karyawan”:

```
$ db.karyawan.drop();
```

Purwadhika
Startup and Coding School

Add Data Record

- Add 1 data JSON ke collection ”karyawan”:

```
$ db.karyawan.insert({nama:"Adi",usia:24});
```

- Add multiple data ke collection “karyawan”:

```
$ db.karyawan.insert([
{nama:"Budi",usia:23},
{nama:"Caca",usia:25}]);
```

- Menampilkan data pada collection “karyawan”:

```
$ db.karyawan.find();
$ db.karyawan.find().pretty();
$ db.karyawan.find({nama: "Adi"}).pretty();
```

mongoDB

Update

- Update **semua** property data:

```
$ db.karyawan.update({nama: "Adi"},
  {nama:"Adi",usia:21});
```

- Update property data tertentu:

```
$ db.karyawan.update({nama: "Budi"},
  {$set:{usia:26}});
```

- Update nama property:

```
$ db.karyawan.update({nama: "Caca"},
  {$rename:{"usia":"umur"}});
```

Purwadhika
Startup and Coding School

mongoDB

Remove

- Hapus 1 property dari sebuah data:

```
$ db.karyawan.update({nama: "Adi"},
{$unset:{usia:true}});

$ db.karyawan.update({nama: "Budi"},
{$unset:{usia:1}});
```

- Hapus 1 data:

```
$ db.karyawan.remove({nama: "Caca"});
```

Purwadhika
Startup and Coding School

mongoDB

And & Or

- Tampilkan data yang memiliki property value nama = Budi ***dan*** usia = 21:

```
$ db.karyawan.find({$and:
[{nama:"Budi"},{usia:21}]}).pretty();
```

- Tampilkan data yang memiliki property value nama = Adi ***atau*** nama = Budi:

```
$ db.karyawan.find({$or:
[{nama:"Adi"},{nama:"Budi"}]}).pretty();
```

Purwadhika
Startup and Coding School

mongoDB

Lower and Greater Than

- Tampilkan data yang property usianya < 25:

```
$ db.karyawan.find({usia:{$lt:25}})
.pretty();
```

- Tampilkan data yang property usianya > 25:

```
$ db.karyawan.find({usia:{$gt:25}})
.pretty();
```

Purwadhika
Startup and Coding School

mongoDB

Lower and Greater Than Equal

- Tampilkan data yang property usianya <= 25:

```
$ db.karyawan.find({usia:{$lte:25}})
.pretty();
```

- Tampilkan data yang property usianya >= 25:

```
$ db.karyawan.find({usia:{$gte:25}})
.pretty();
```

Purwadhika
Startup and Coding School

mongoDB

Limit & Skip

- Tampilkan 2 data pertama:

```
$ db.karyawan.find().limit(2);
```

- Tampilkan 1 data setelah 2 data pertama:

```
$ db.karyawan.find().limit(1).skip(2);
```

- Tampilkan 3 data setelah 3 data pertama:

```
$ db.karyawan.find().limit(3).skip(3);
```

Purwadhika
Startup and Coding School

mongoDB

Sort

- Urutkan data ***ascending*** berdasarkan ***nama***:

```
$ db.karyawan.find().sort({nama:1});
```

- Urutkan data ***descending*** berdasarkan ***nama***:

```
$ db.karyawan.find().sort({nama:-1});
```

- Urutkan data ***ascending*** berdasarkan ***usia***:

```
$ db.karyawan.find().sort({usia:1});
```

- Urutkan data ***descending*** berdasarkan ***usia***:

```
$ db.karyawan.find().sort({usia:-1});
```

Purwadhika
Startup and Coding School

mongoDB

Count

- Hitung jumlah data di collection “karyawan”:

```
$ db.karyawan.find().count();
```

- Hitung jumlah data dg prop nama = Adi:

```
$ db.karyawan.find({nama:"Adi"}).count();
```

Purwadhika
Startup and Coding School

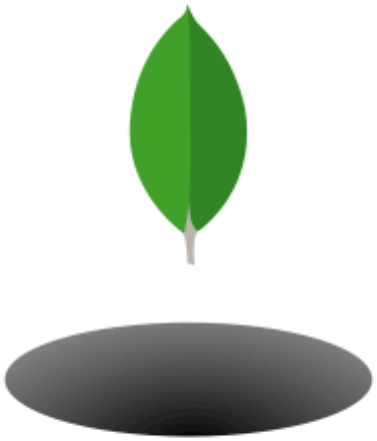
How to Work With MongoDB GUI Tools

Purwadhika
Startup and Coding School

mongoDB

Working with GUI

#1 Installing MongoDB Compass



MongoDB Compass is being installed.

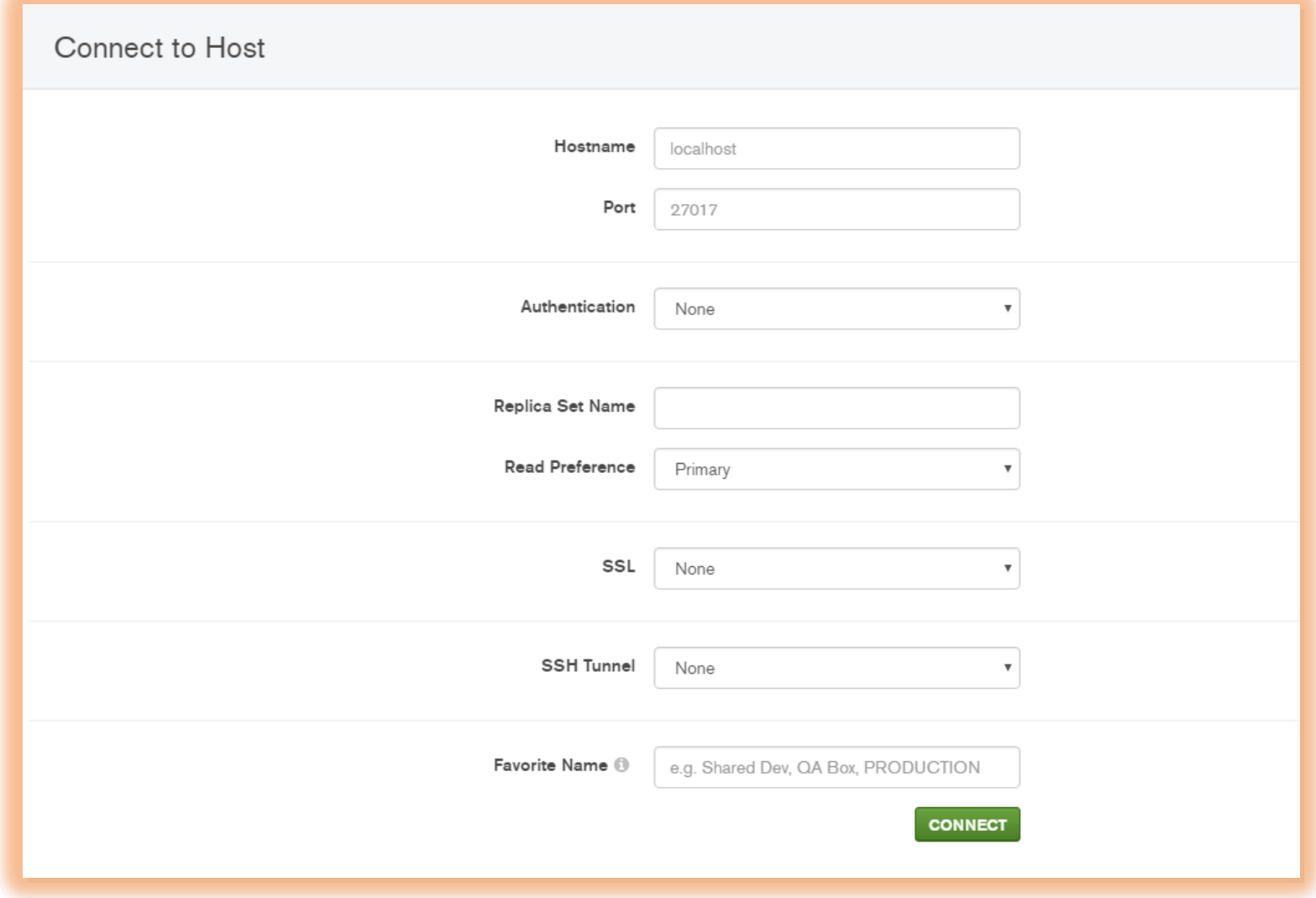
It will launch once it is done.

Purwadhika
Startup and Coding School

mongoDB

Working with GUI

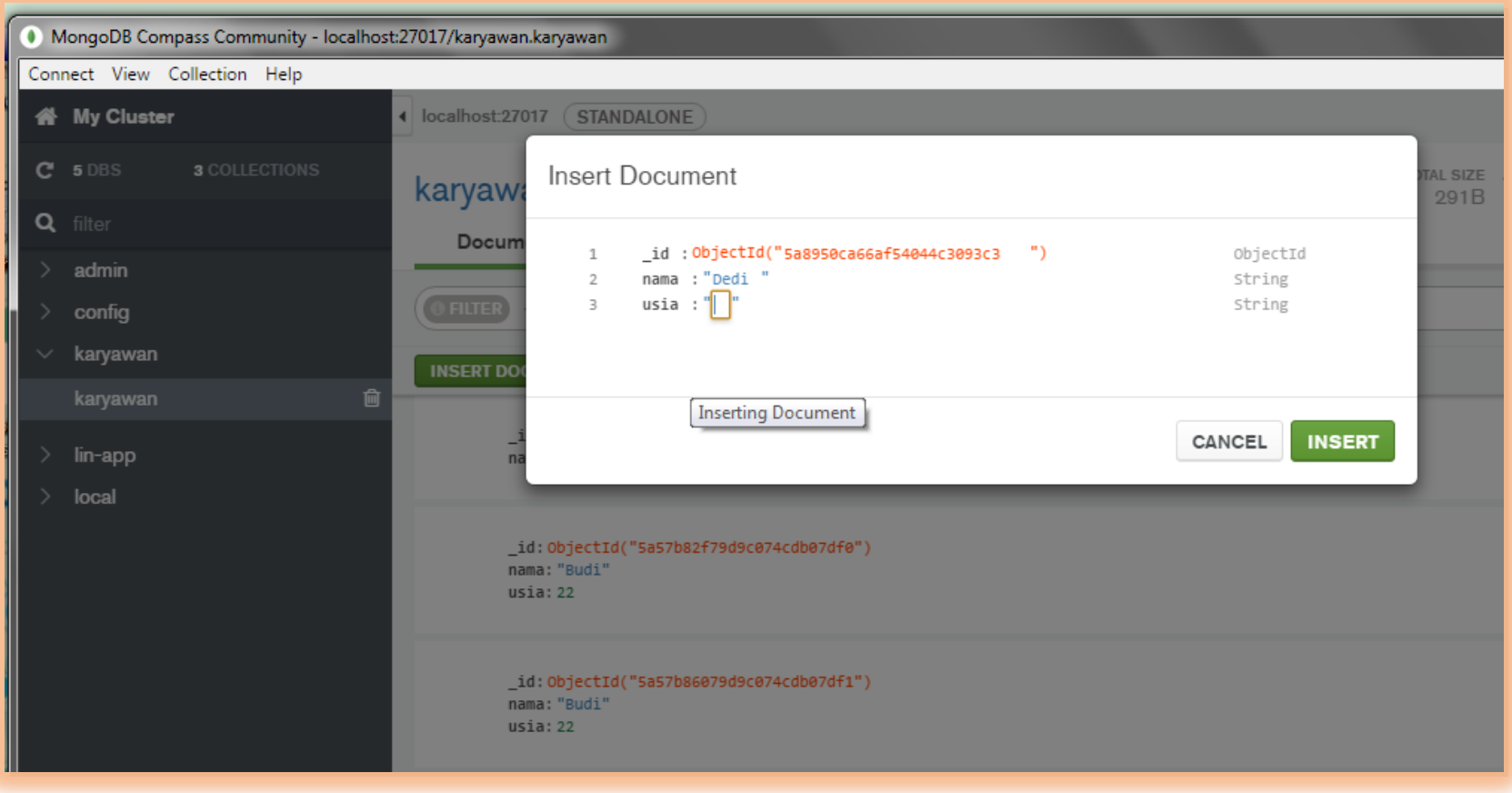
#2 Connect to Server



mongoDB

Working with GUI

#3 Explore by yourself!

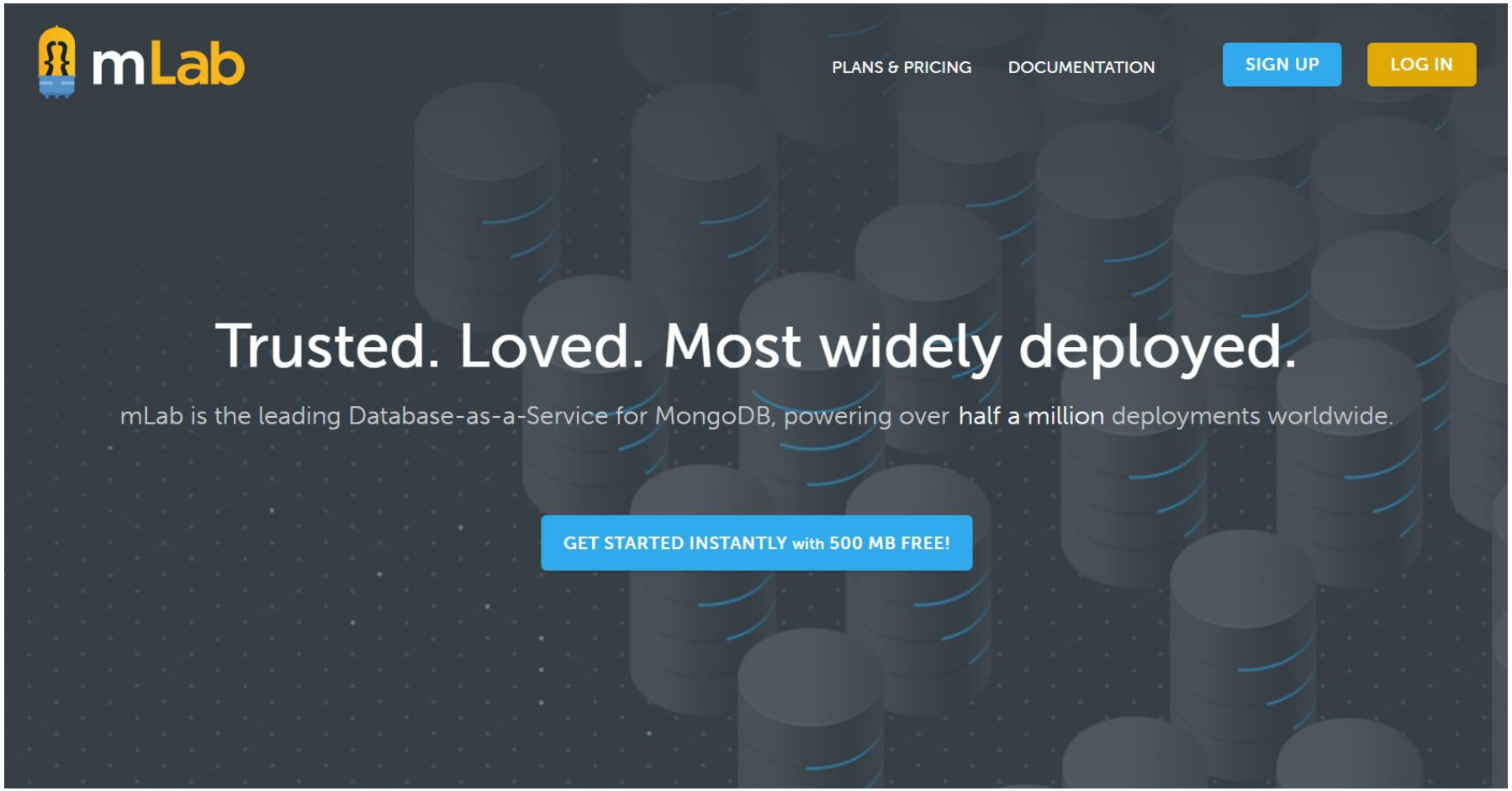


Purwadhika
Startup and Coding School

How to Work With MongoDB Hosting

Purwadhika
Startup and Coding School

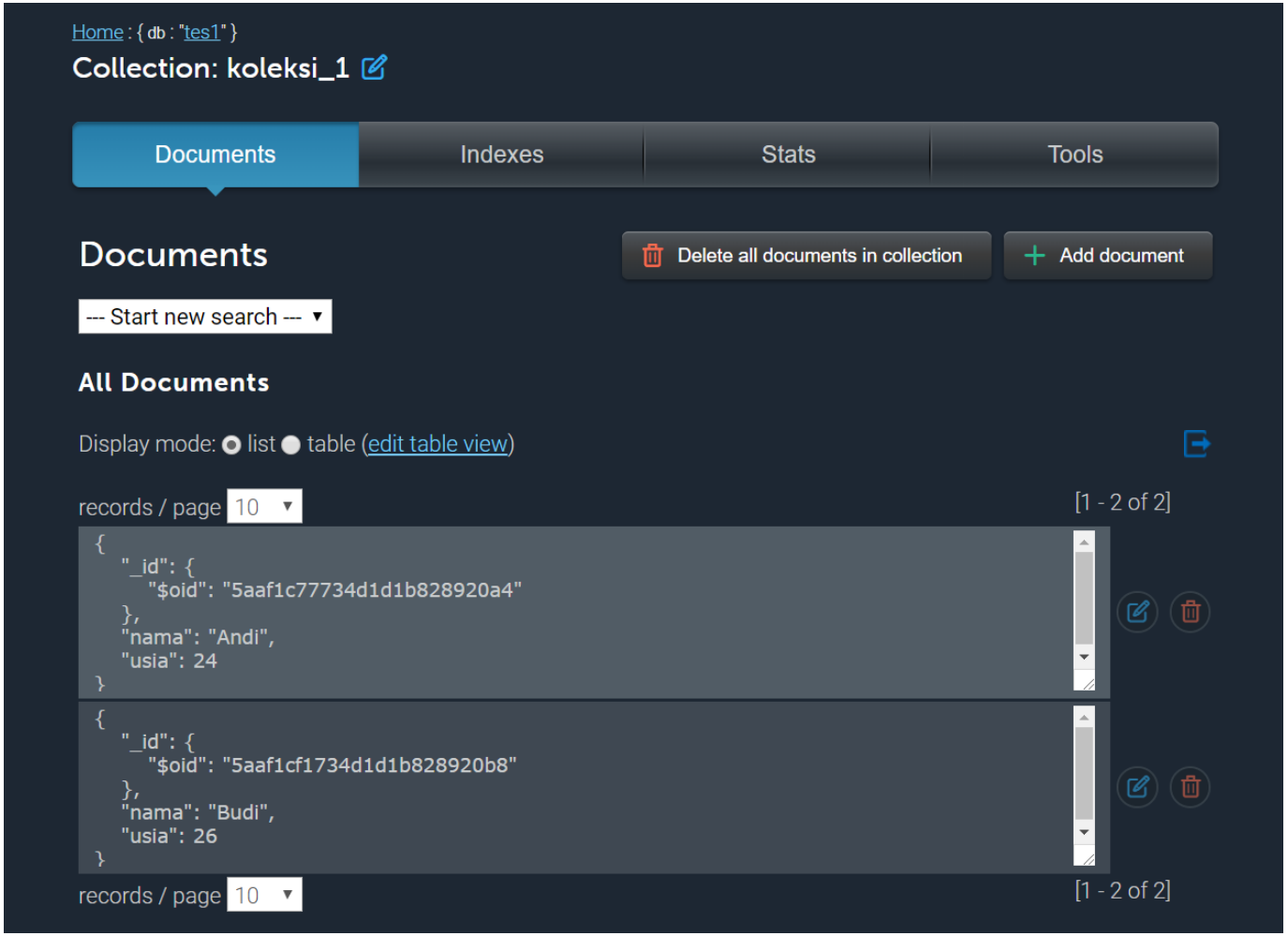
mLab



mlab.com/

Purwadhika
Startup and Coding School

mLab



mlab.com/

Purwadhika
Startup and Coding School

mLab

Create document

```
{
    "nama": "Caca",
    "usia": 23
}
```

cancel and go back | Create and go back | Create and continue editing

mlab.com/

Purwadhika
Startup and Coding School

Back-End Development

Exploration

Purwadhika
Startup and Coding School