

*Berdo'alah sebelum mengerjakan. Dilarang berbuat curang.  
Tugas ini untuk mengukur kemampuan anda, jadi kerjakan dengan sepenuh hati.  
Selamat belajar, semoga sukses !*

Nama Mahasiswa: Yuwantoro Mukhlisin .....	NIM: 1301150042 .....	Nilai: .....
Nama Mahasiswa: Lukman Budiman .....	NIM: 1301164725 .....	Nilai: .....
Nama Mahasiswa: Salma Fauzia Susan .....	NIM: 1301164442 .....	Nilai: .....

**Siapkan tools berikut sebelum mengerjakan:**

1. Go Programming Language (<https://golang.org/dl/>).
2. Visual Studio Code (<https://code.visualstudio.com/>) atau LiteIDE (<https://github.com/visualfc/liteide>).
3. Harus menggunakan linux dengan distro fedora (<https://getfedora.org/id/workstation/>).
4. Pakailah Mosquitto MQTT untuk tugas ini (<https://mosquitto.org/>).
5. Gunakan database MongoDB (<https://www.mongodb.com/>) untuk membuat aplikasi CRUD.
6. Buatlah git repository pada <https://github.com/> kemudian push semua kode dan hasil laporan anda ke dalam repository github yang sudah anda buat.
7. Kumpulkan link repository github tersebut sebagai tanda bahwa anda mengerjakan tugas modul ini.
8. Link repository harus berbeda untuk setiap tugasnya. Buatlah markdown yang rapi di setiap repository tugas yang anda kumpulkan.
9. Printscreen program harus dari desktop kelompok anda sendiri, dan harus dari linux yang sudah diinstall. Jika tidak, maka harus mengulang pengerjaan tugasnya.
10. Jangan lupa untuk menuliskan NAMA dan NIM pada laporan.
11. Laporan berbentuk PDF dan dikumpulkan pada link repository github beserta kodenya.
12. Walaupun tugas berkelompok tapi pengumpulan link github harus individu, jika tidak mengumpulkan maka dianggap tidak mengerjakan.

Nama:	NIM:	Nilai:
-------	------	--------

### Soal No 1 (HTTP Web Application CRUD)

Buatlah sebuah web aplikasi (harus ada tampilan di browser) yang dapat digunakan untuk menginputkan, melihat, memperbaharui, dan menghapus data karyawan (nama, email, nomor telpon, dan alamat). Gunakan database MongoDB untuk penyimpanan datanya.

Gunakan library <https://github.com/gorilla/mux> untuk membuat web aplikasi anda.

Jawaban:

```
func create(c *mongo.Collection) {
    err := c.Insert(
        &Person{Name: "Eli", Phone: "111-239-333", Timestamp: time.Now()},
        &Person{Name: "Ben", Phone: "111-239-331", Timestamp: time.Now()},
        &Person{Name: "Jun", Phone: "111-239-222", Timestamp: time.Now()},
        &Person{Name: "Len", Phone: "111-231-333", Timestamp: time.Now()},
    )

    if err != nil {
        fmt.Println(err)
    }

    fmt.Printf("\n\n")
}
```

```
func read(c *mongo.Collection) {
    // Read Once
    result := Person{}
    query := c.Find(bson.M{"name": "Eli"})
    query = query.Select(bson.M{"phone": 0})
    err := query.One(&result)

    if err != nil {
        fmt.Println(err)
        return
    }

    fmt.Println(result)

    // Read alot
    var results []Person
    query = c.Find(bson.M{}).Sort("-timestamp")
    err = query.All(&results)

    for _, value := range results {
        fmt.Println(value)
    }

    fmt.Printf("\n\n")
}
```

Nama:	NIM:	Nilai:
-------	------	--------

```
func update(c *mgo.Collection) {  
  
    filter := bson.M{"name": "Eli"}  
    change := bson.M{  
        "$set": bson.M{  
            "phone": "+86 99 8888 7707",  
            "timestamp": time.Now(),  
        },  
    }  
  
    err := c.Update(filter, change)  
    if err != nil {  
        fmt.Println(err)  
        return  
    }  
  
    read(c)  
    fmt.Printf("\n\n")  
}
```

```
func delete(c *mgo.Collection) {  
  
    filter := bson.M{"name": "Ben"}  
    err := c.Remove(filter)  
  
    if err != nil {  
        fmt.Println(err)  
        return  
    }  
  
    read(c)  
    fmt.Printf("\n\n")  
}
```

Hasil :

Nama:	NIM:	Nilai:
-------	------	--------

### 1. Read

```
{Eli 2019-09-20 14:15:43.657 +0700 WIB}
{Eli 111-239-333 2019-09-20 14:15:43.657 +0700 WIB}
{Ben 111-239-331 2019-09-20 14:15:43.657 +0700 WIB}
{Jun 111-239-222 2019-09-20 14:15:43.657 +0700 WIB}
{Len 111-231-333 2019-09-20 14:15:43.657 +0700 WIB}
```

### 2. Update

```
{Eli 2019-09-20 14:15:43.659 +0700 WIB}
{Eli +86 99 8888 7707 2019-09-20 14:15:43.659 +0700 WIB}
{Ben 111-239-331 2019-09-20 14:15:43.657 +0700 WIB}
{Jun 111-239-222 2019-09-20 14:15:43.657 +0700 WIB}
{Len 111-231-333 2019-09-20 14:15:43.657 +0700 WIB}
```

no telp eli berubah

### 3. Delete

```
{Eli 2019-09-20 14:15:43.659 +0700 WIB}
{Eli +86 99 8888 7707 2019-09-20 14:15:43.659 +0700 WIB}
{Jun 111-239-222 2019-09-20 14:15:43.657 +0700 WIB}
{Len 111-231-333 2019-09-20 14:15:43.657 +0700 WIB}
```

3.

nama ben dihapus.

**Soal No 2 (MQTT in Golang)**

1. Buatlah sebuah aplikasi yang bisa melakukan publish kepada MQTT broker.
2. Buatlah sebuah aplikasi yang bisa melakukan subscribe kepada MQTT broker.
3. Gunakan library ini untuk membuat aplikasi anda  
<https://github.com/eclipse/paho.mqtt.golang>

Jawaban: