

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

Nama Mahasiswa: I Putu Surya Baratha	NIM: 1301188566	Nilai:
Nama Mahasiswa: M. Risdham Nur Ade P.	NIM: 1301188603	Nilai:
Nama Mahasiswa: Sella Tresnasari	NIM: 1301188565	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:
-------	------	--------

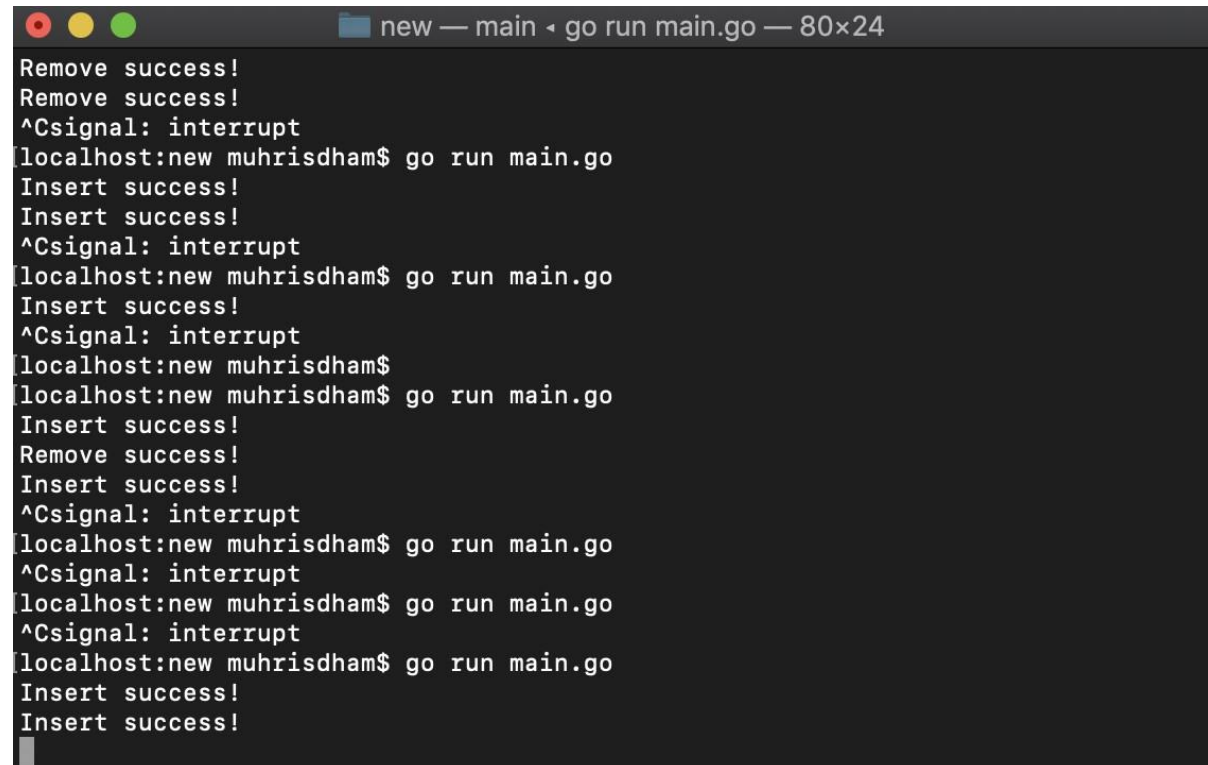
Soal No 1 (HTTP Web Application CRUD)

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:

Database yang dipakai adalah MongoDB. MongoDB adalah sistem basis data berorientasi dokumen lintas platform. MongoDB diklasifikasikan sebagai basis data yang “NoSQL”, MongoDB menghindari struktur basis data relasional tabel berbasis tradisional yang mendukung JSON seperti dokumen dengan skema dinamis, membuat integrase data dalam beberapa jenis aplikasi lebih mudah dan juga cepat.



```

new — main ◀ go run main.go — 80x24
Remove success!
Remove success!
^Csignal: interrupt
localhost:new muhrisdham$ go run main.go
Insert success!
Insert success!
^Csignal: interrupt
localhost:new muhrisdham$ go run main.go
Insert success!
^Csignal: interrupt
localhost:new muhrisdham$
localhost:new muhrisdham$ go run main.go
Insert success!
Remove success!
Insert success!
^Csignal: interrupt
localhost:new muhrisdham$ go run main.go
^Csignal: interrupt
localhost:new muhrisdham$ go run main.go
^Csignal: interrupt
localhost:new muhrisdham$ go run main.go
Insert success!
Insert success!

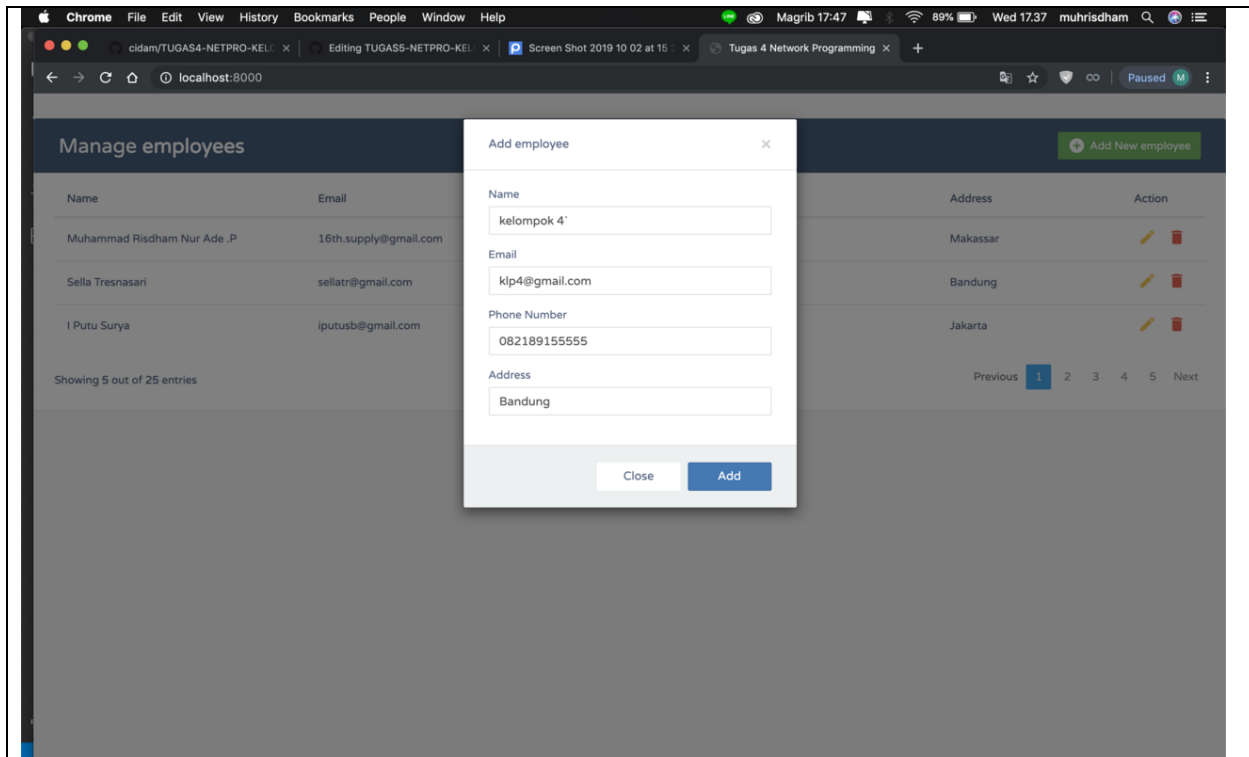
```

Gambar program 1

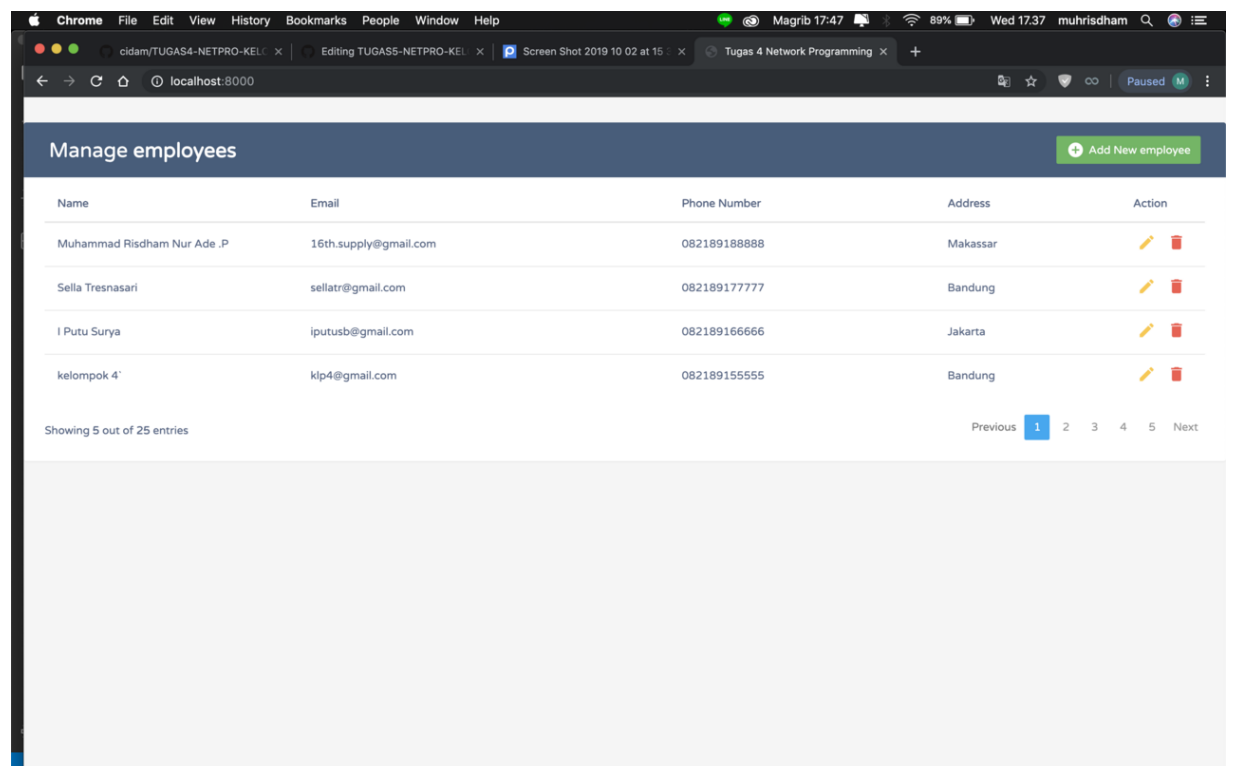
Nama:

NIM:

Nilai:



Gambar program 2

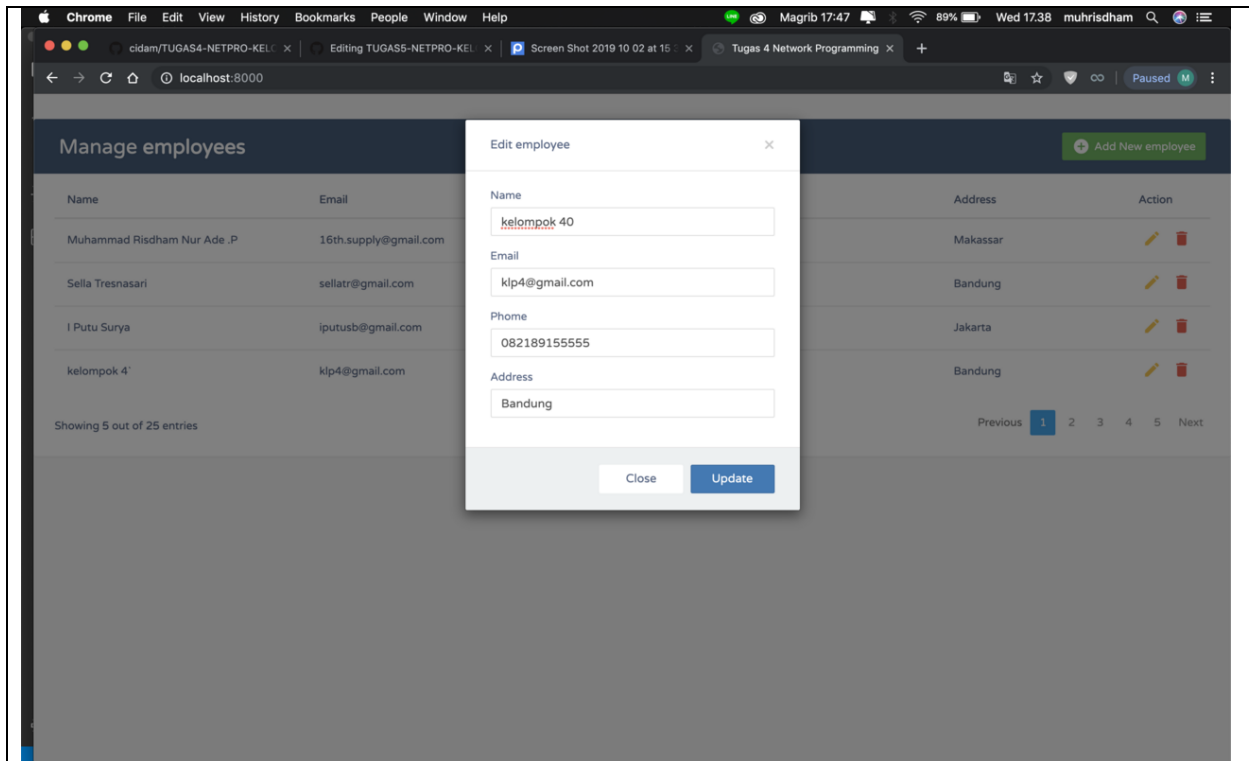


Gambar program 3

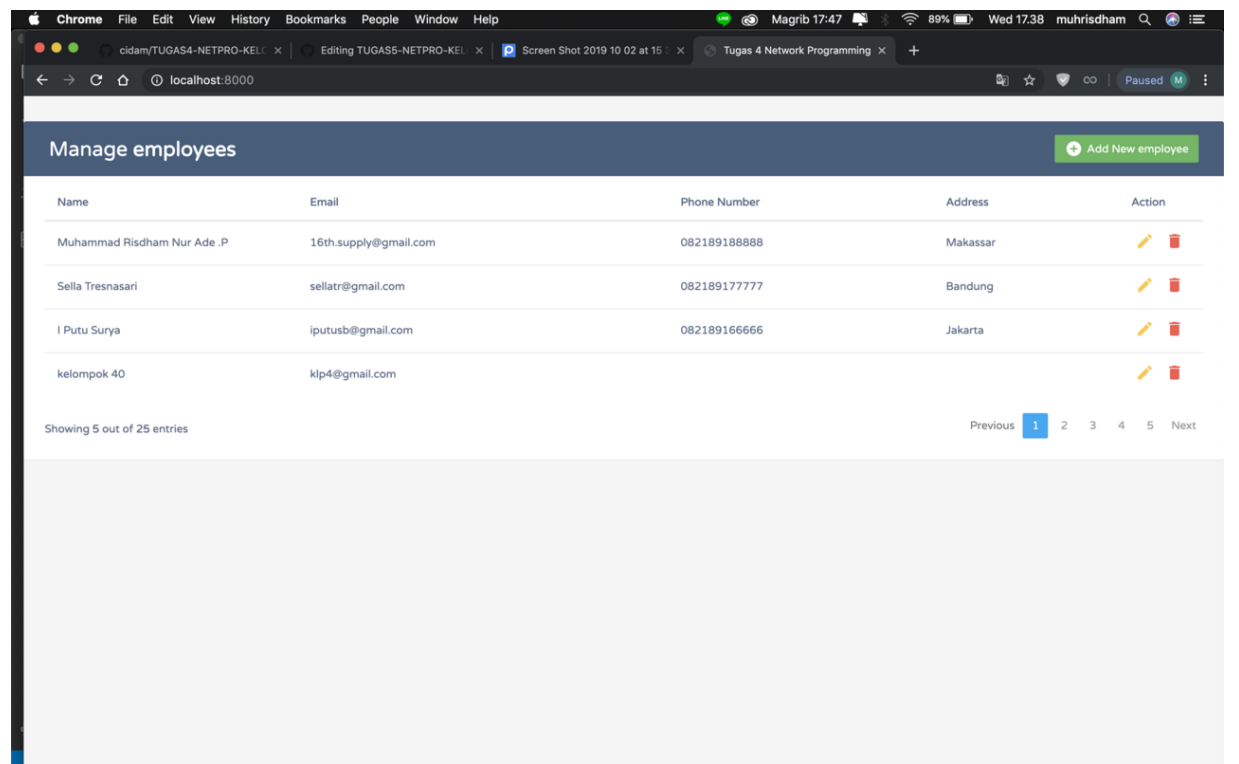
Nama:

NIM:

Nilai:



Gambar program 4

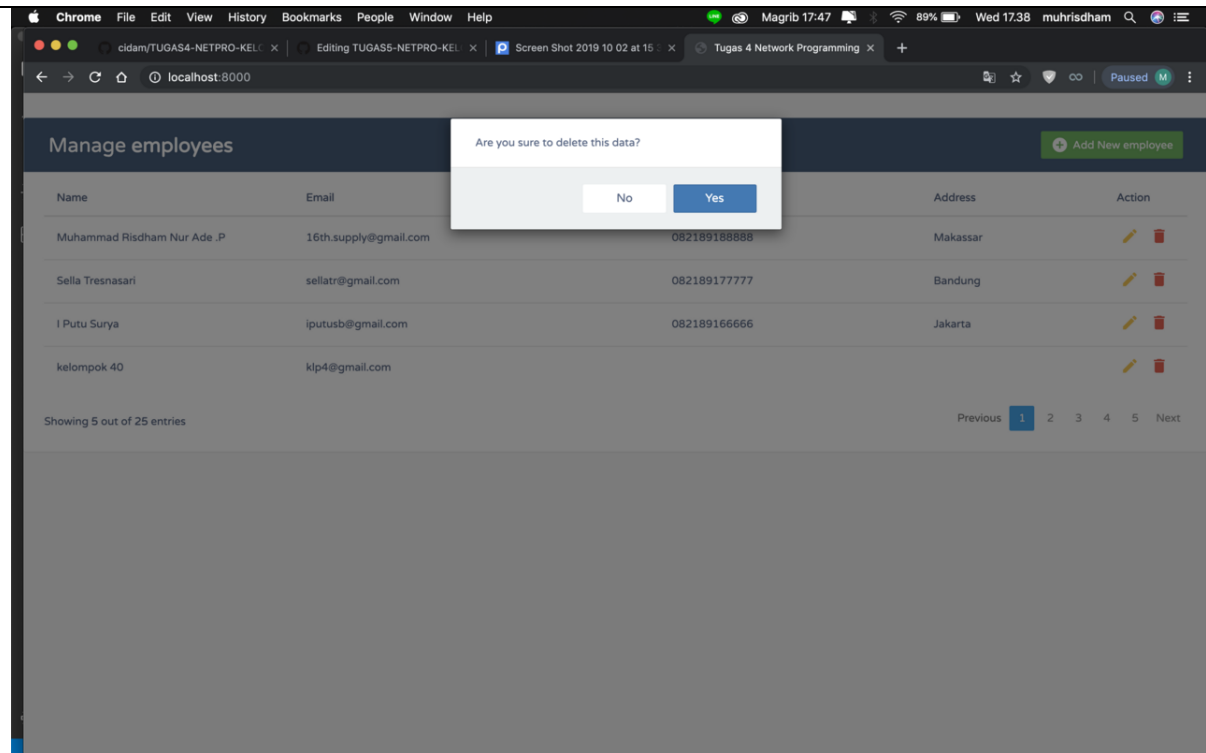


Gambar program 5

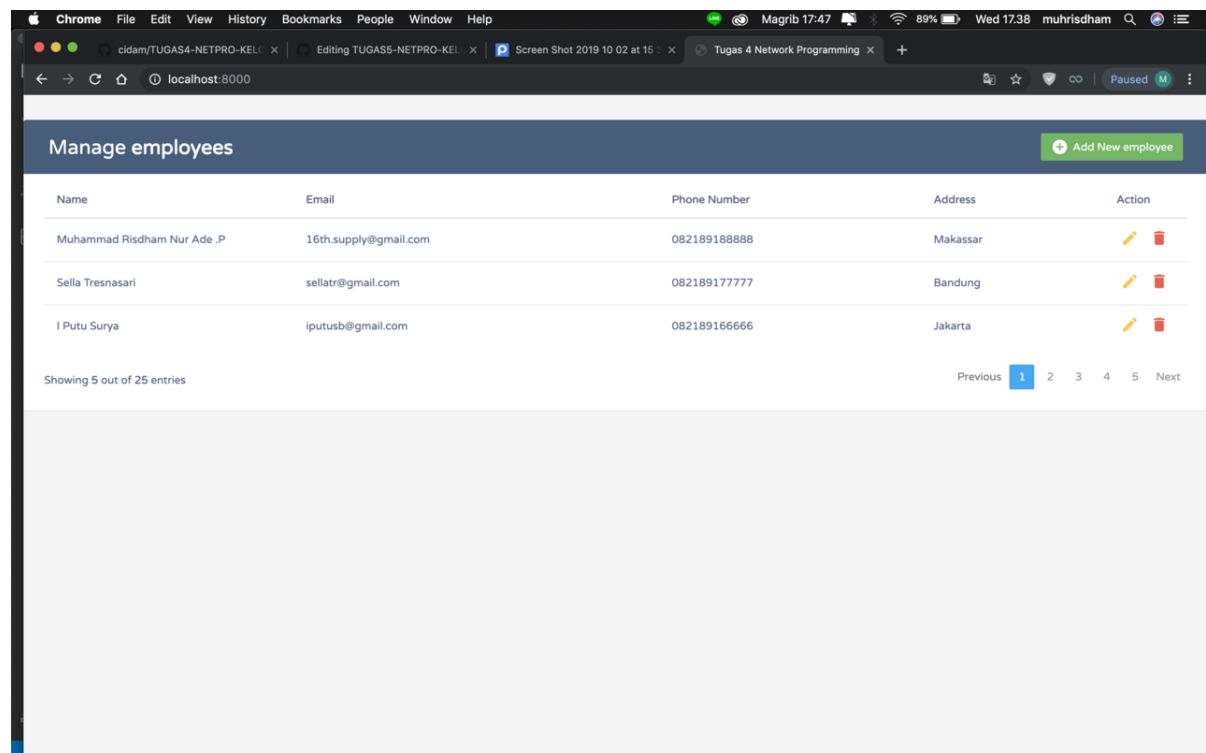
Nama:

NIM:

Nilai:



Gambar program 6



Gambar program 7

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

Keterangan gambar:

- Gambar 1 : Pada gambar pertama, fungsinya untuk memberikan pemberitahuan ketika kita melakukan insert, update maupun delete pada data berhasil atau tidak.
- Gambar 2 : Pada gambar kedua ini berfungsi untuk menambahkan data karyawan baru dengan format nama, email, phone number, dan address
- Gambar 3 : Pada gambar ketiga ini adalah hasil dari penambahan data karyawan baru pada gambar sebelumnya
- Gambar 4 : Pada gambar keempat berfungsi untuk mengedit data karyawan. Data-data yang dapat diedit adalah nama, email, phone number dan juga address
- Gambar 5 : Pada gambar kelima adalah hasil edit pada gambar di nomor 4. Hasil editannya adalah namanya berubah dari yang sebelumnya kelompok 4 menjadi kelompok 40
- Gambar 6 : Pada gambar keenam berfungsi untuk menghapus data karyawan, tapi sebelum akan dihapus akan muncul konfirmasi untuk menanyakan kembali apakah benar-benar akan dihapus data karyawannya
- Gambar 7 : Pada gambar terakhir adalah hasil setelah data karyawan kelompok 40 dihapus pada gambar sebelumnya

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

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:

The screenshot shows a Go IDE with three windows. The main window displays the source code for two Go programs: 'pubs.go' and 'subs.go'. The 'pubs.go' code defines a publisher that connects to an MQTT broker, publishes a message 'Tugas 4 Network Programming' to the topic 'TUGAS 4', and then disconnects. The 'subs.go' code defines a subscriber that connects to the same MQTT broker, subscribes to the topic 'TUGAS 4', and prints the received message. Two terminal windows are also shown. The first terminal window shows the output of running 'subs.go', which prints 'Topic: TUGAS 4' and 'Message: Tugas 4 Network Programming'. The second terminal window shows the output of running 'pubs.go', which prints 'Topic: TUGAS 4' and 'Message: Tugas 4 Network Programming'.

```

src > TUGAS4 > pubs.go > {}main > main
1 package main
2
3 import (
4     "fmt"
5
6     MQTT "github.com/eclipse/paho
7 )
8
9 var f MQTT.MessageHandler = func(
10     fmt.Printf("Topic: %s\n", msg
11     fmt.Printf("Message: %s\n", m
12 )
13
14 func main() {
15
16     opts := MQTT.NewClientOptions
17     opts.SetClientID("publisher")
18     opts.SetDefaultPublishHandler
19
20
21     c := MQTT.NewClient(opts)
22     if token := c.Connect(); token.Wait() && token.Error() !=
23         panic(token.Error())
24 }
25
26 msg := fmt.Sprintf("Tugas 4 Network Programming")
27 token := c.Publish("TUGAS 4", 0, false, msg)
28 token.Wait()
29
30 c.Disconnect(250)
31 }
32
TUGAS4 -- subs - go run subs.go -- 80x24
localhost:tugas4 muhrisdham$ go run subs.go
Topic: TUGAS 4
Message: Tugas 4 Network Programming

TUGAS4 -- -bash -- 80x24
localhost:TUGAS4 muhrisdham$ go run pubs.go
localhost:TUGAS4 muhrisdham$ go run pubs.go

```

MQTT (Message Queuing Telemetry Transport) merupakan suatu protokol yang biasa digunakan untuk berkomunikasi lewat internet. Beberapa istilah penting dalam MQTT adalah

- Publish dan Subscribe
- Topic
- Messages
- BrokerPublish dan Subscribe

MQTT menggunakan model Publish Subscribe. Publisher yang dapat melakukan Publish yaitu membagikan suatu "topic" dan Subscriber yang dapat melakukan subscribe yaitu menerima suatu "topic". Jadi, Publish dan Topic ini seperti tipe data khusus untuk membagikan atau menerima suatu topic. Jadi dalam penggunaannya pada protokol MQTT, akan dideklarasikan terlebih dahulu suatu variabel misalkan

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

```
func
main()
{
    // Config untuk rest3e
    opts := MQTT.NewClientOptions().AddBroker("localhost:1883")
    opts.SetClientID("subscriber")
    opts.SetDefaultPublishHandler(f)
}

dan

func
main()
{
    // Config
    opts := MQTT.NewClientOptions().AddBroker("localhost:1883")
    opts.SetClientID("publisher")
    opts.SetDefaultPublishHandler(f)
}
```