# LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : ALGORITMA PEMOGRAMAN 3B

Kelas : 2IA11

Praktikum ke- : 3

Tanggal : 4 Januari 2024

Materi : KONSEP DAN KONEKSI BASIS DATA DI GOLANG

NPM : 51422161

Nama : MUHAMMAD TARMIDZI BARIQ

Ketua Asisten : SRI MEINIS

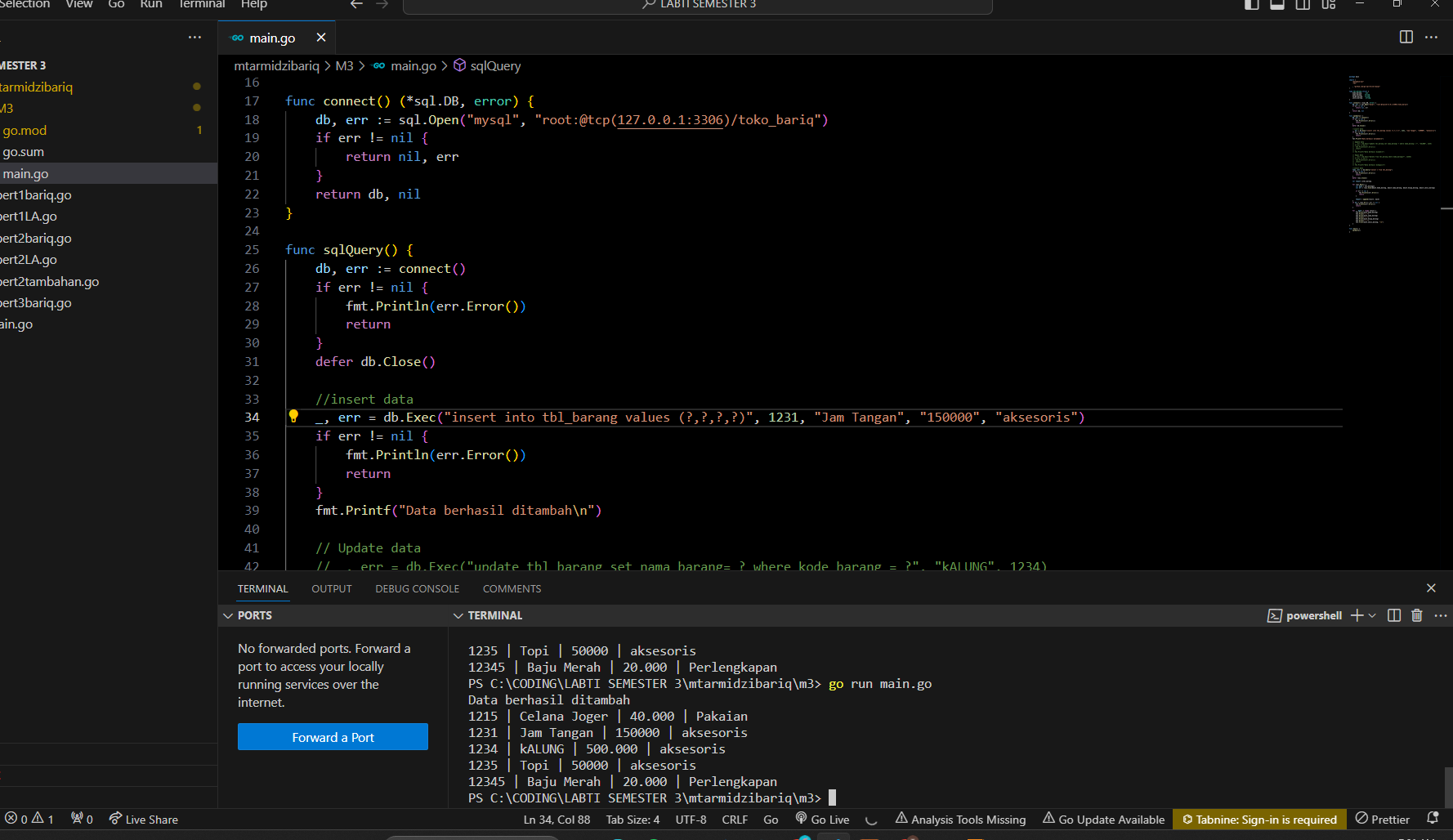
Jumlah Lembar : 3



# LABORATORIUM TEKNIK INFORMATIKA UNIVERSITAS GUNADARMA

**2024**

1. system yang dirancang untuk menyimpan, menghapus. mengupdate, mengelola data secara terstruktur aplikasi yang digunakan bisa XAMPP, Firebase, Oracle, dll. Terdapat relasi yang digunakan untuk menghubungkan beberapa table menggunakan primary key dan foreign key

**2. **



package main

import (

    "database/sql"

    "fmt"

    \_ "github.com/go-sql-driver/mysql"

)

type tbl\_barang struct {

    kode\_barang     int

    nama\_barang    string

    harga\_barang   string

    jenis\_barang    string

}

func connect() (\*sql.DB, error) {

    db, err := sql.Open("mysql", "root:@tcp(127.0.0.1:3306)/toko\_bariq")

    if err != nil {

        return nil, err

    }

    return db, nil

}

func sqlQuery() {

    db, err := connect()

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    defer db.Close()

    //insert data

    \_, err = db.Exec("insert into tbl\_barang values (?,?,?,?)", 1231, "Jam Tangan", "150000", "aksesoris")

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    fmt.Printf("Data berhasil ditambah\n")

    // Update data

    // \_, err = db.Exec("update tbl\_barang set nama\_barang= ? where kode\_barang = ?", "kALUNG", 1234)

    // if err != nil {

    //  fmt.Println(err.Error())

    //  return

    // }

    // fmt.Printf("Data berhasil diubah\n")

    // Hapus Data

    // \_, err = db.Exec("delete from tbl\_barang where kode\_barang=?", 12345)

    // if err != nil {

    //  fmt.Println(err.Error())

    //  return

    // }

    // fmt.Printf("Data berhasil dihapus\n")

    //select data

    rows, err := db.Query("select \* from tbl\_barang")

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    defer rows.Close()

    var result []tbl\_barang

    for rows.Next() {

        var each = tbl\_barang{}

        var err = rows.Scan(&each.kode\_barang, &each.nama\_barang, &each.harga\_barang, &each.jenis\_barang)

        if err != nil {

            fmt.Println(err.Error())

            return

        }

        result = append(result, each)

    }

    if err = rows.Err(); err != nil {

        fmt.Println(err.Error())

        return

    }

    for \_, each := range result {

        fmt.Print(each.kode\_barang)

        fmt.Print(" | ")

        fmt.Print(each.nama\_barang)

        fmt.Print(" | ")

        fmt.Print(each.harga\_barang)

        fmt.Print(" | ")

        fmt.Print(each.jenis\_barang, "\n")

    }

}

func main() {

    sqlQuery()

}

Import Packages:

database/sql: Paket standar Go untuk interaksi dengan database SQL.

\_ "github.com/go-sql-driver/mysql": Mengimpor driver MySQL secara anonim untuk digunakan oleh paket database/sql. Garis bawah (\_) mengindikasikan bahwa hanya inisialisasinya yang dibutuhkan, tanpa memerlukan akses langsung ke fungsi atau variabel dalam paket tersebut.

Definisi Struktur Data (tbl\_barang):

Struktur tbl\_barang mendefinisikan struktur data untuk tabel barang dengan kolom kode\_barang, nama\_barang, harga\_barang, dan jenis\_barang.

Fungsi Connect:

connect(): Fungsi ini membuka koneksi ke database MySQL menggunakan kredensial yang disediakan. Alamat database adalah localhost (127.0.0.1) pada port 3306, dengan nama database toko\_bariq.

Fungsi sqlQuery:

Fungsi ini bertanggung jawab untuk menjalankan operasi CRUD.

Insert Data: Menambahkan data baru ke dalam tabel tbl\_barang.

Update Data: Memperbarui nama\_barang di tabel tbl\_barang berdasarkan kode\_barang.

Delete Data: Menghapus data dari tabel tbl\_barang berdasarkan kode\_barang.

Select Data: Mengambil semua data dari tabel tbl\_barang dan menampilkannya.

Setiap operasi diikuti dengan penanganan error dan pesan konfirmasi jika operasi berhasil.

Fungsi main:

Fungsi main adalah titik masuk program dan hanya memanggil fungsi sqlQuery.

Penanganan Error dan Penutupan Koneksi:

Setiap panggilan ke database diikuti dengan penanganan error untuk memastikan program tidak berhenti jika terjadi kesalahan.

Koneksi ke database ditutup (defer db.Close()) setelah operasi selesai untuk menghindari kebocoran sumber daya.

Iterasi dan Tampilan Hasil Query:

Hasil dari query select diiterasi, dan setiap baris hasil diambil datanya, kemudian ditambahkan ke slice result.

Setelah itu, hasil query ditampilkan ke konsol.

Secara keseluruhan, program ini mendemonstrasikan penggunaan dasar dari operasi database dalam Go dengan koneksi ke MySQL, menampilkan penerapan CRUD dalam konteks pengelolaan data barang.