**LAPORAN PRAKTIKUM LATIHAN**

Pemrograman Berbasis Web

Pertemuan 11



Oleh :

NAMA : FARAH TRI MAHARDINI

NPM : 4522210042

Dosen:

**Adi Wahyu Pribadi ,S.Si.,M.Kom**

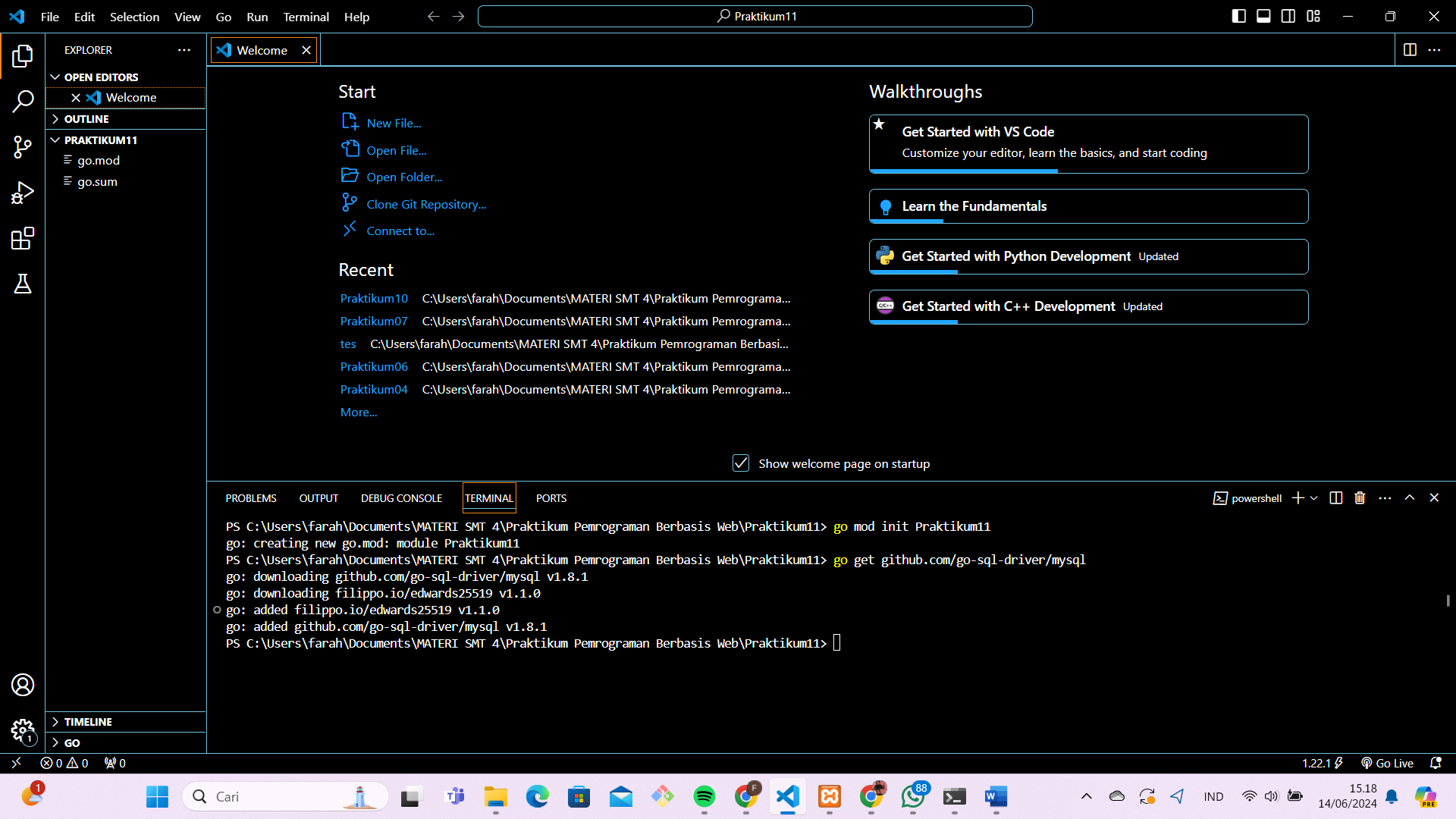
**S1-Teknik Informatika**

**Fakultas Teknik Universitas Pancasila**

**2023/2024**

**LINK GITHUB :**

**56.1 Intalasi Driver**



**56.2 Setup Database**

-- Database: `farah\_belajar\_golang`

-- Table structure for table `farah\_student`

CREATE TABLE IF NOT EXISTS `farah\_student` (

  `id` varchar(5) NOT NULL,

  `name` varchar(255) NOT NULL,

  `age` int(11) NOT NULL,

  `grade` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `farah\_student` (`id`, `name`, `age`, `grade`) VALUES

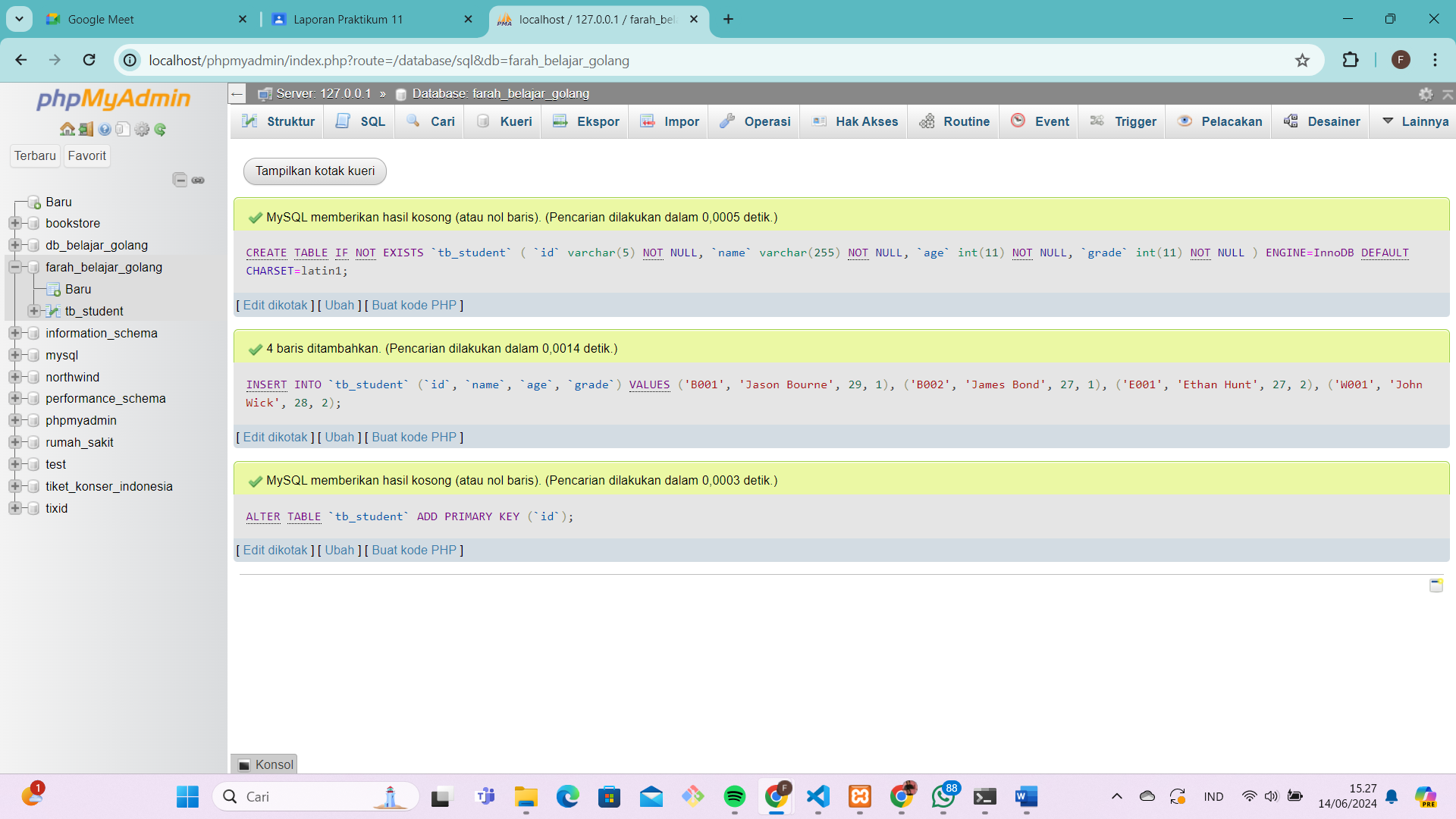
('B001', 'Jason Bourne', 29, 1),

('B002', 'James Bond', 27, 1),

('E001', 'Ethan Hunt', 27, 2),

('W001', 'John Wick', 28, 2);

ALTER TABLE `farah\_student` ADD PRIMARY KEY (`id`);



**56.3 Membaca Data Dari MySQL Server**

package main

import "fmt"

import "database/sql"

import "github.com/go-sql-driver/mysql"

type student struct {

    id    string

    name  string

    age   int

    grade int

}

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

    db, err := sql.Open("mysql", "root:@tcp(127.0.0.1:3306)/farah\_belajar\_golang")

    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()

    var age = 27

    rows, err := db.Query("select id, name, grade from tb\_student where age = ?", age)

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    defer rows.Close()

    var result []student

    for rows.Next() {

        var each = student{}

        var err = rows.Scan(&each.id, &each.name, &each.grade)

        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.Println(each.name)

    }

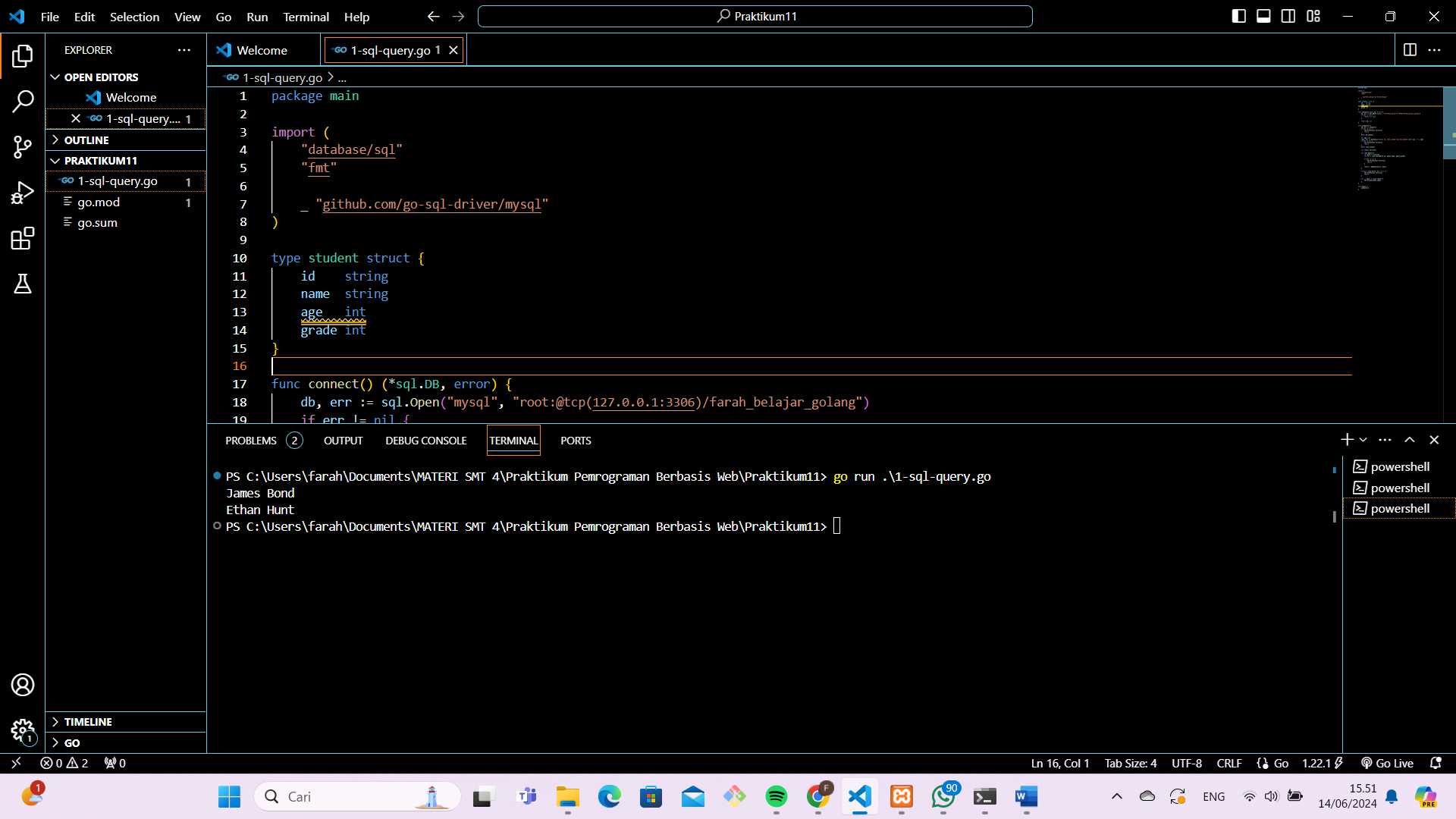
}

func main() {

    sqlQuery()

}

Akan menampilkan output



**56.4 Membaca 1 Recoed Data menggunakan Method**

**QueryRow()**

package main

import (

    "database/sql"

    "fmt"

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

)

type student struct {

    id    string

    name  string

    age   int

    grade int

}

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

    db, err := sql.Open("mysql", "root:@tcp(127.0.0.1:3306)/farah\_belajar\_golang")

    if err != nil {

        return nil, err

    }

    return db, nil

}

func sqlQueryRow() {

    var db, err = connect()

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    defer db.Close()

    var result = student{}

    var id = "E001"

    err = db.

        QueryRow("select name, grade from tb\_student where id = ?", id).

        Scan(&result.name, &result.grade)

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    fmt.Printf("name: %s\ngrade: %d\n", result.name, result.grade)

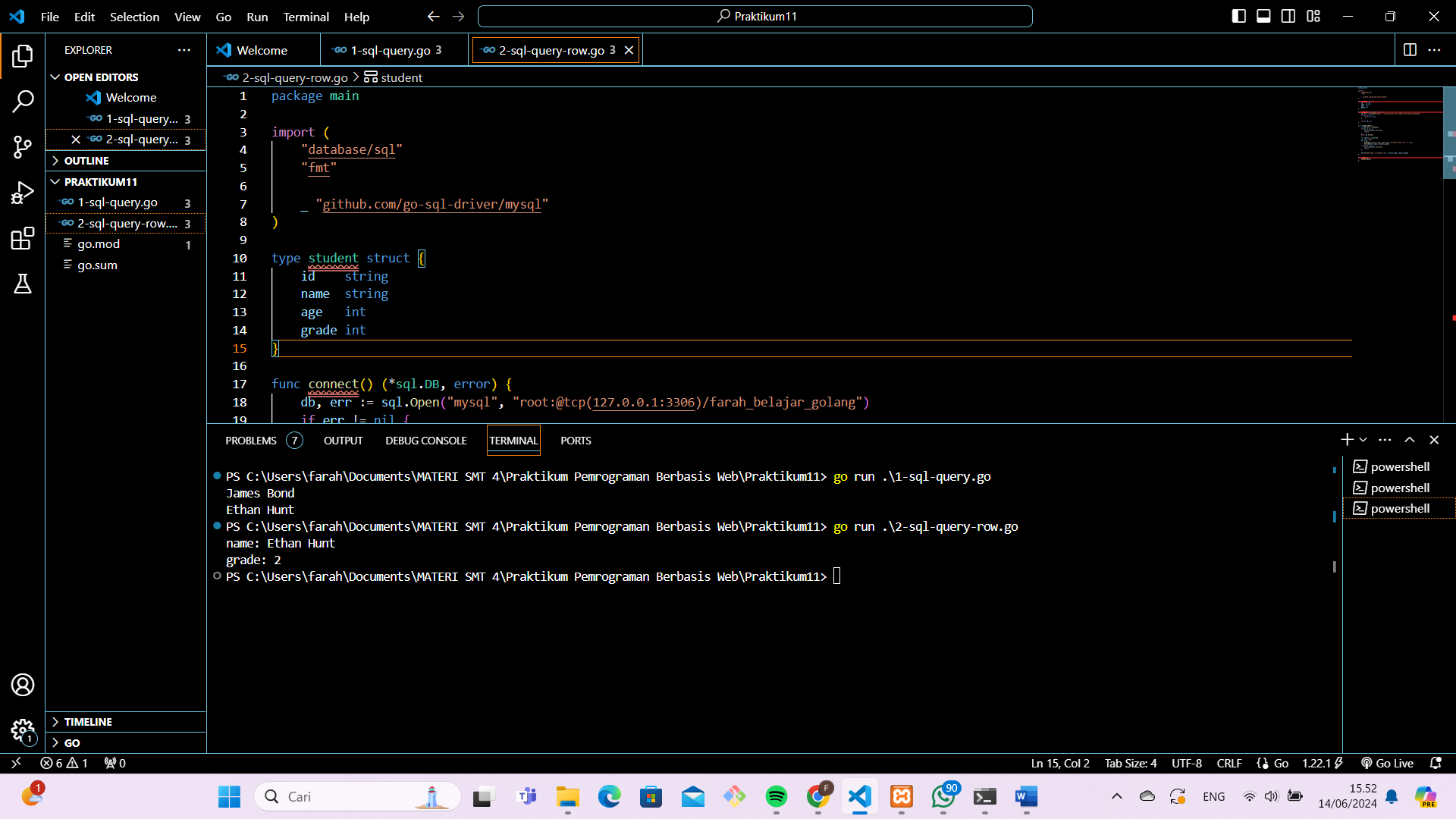
}

func main() {

    sqlQueryRow()

}

Akan menghasilkan output



**56.5 Eksekusi Query Menggunakan**

**Prepare()**

package main

import (

    "database/sql"

    "fmt"

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

)

type student struct {

    id    string

    name  string

    age   int

    grade int

}

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

    db, err := sql.Open("mysql", "root:@tcp(127.0.0.1:3306)/farah\_belajar\_golang")

    if err != nil {

        return nil, err

    }

    return db, nil

}

func sqlPrepare() {

    db, err := connect()

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    defer db.Close()

    stmt, err := db.Prepare("select name, grade from tb\_student where id = ?")

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    var result1 = student{}

    stmt.QueryRow("E001").Scan(&result1.name, &result1.grade)

    fmt.Printf("name: %s\ngrade: %d\n", result1.name, result1.grade)

    var result2 = student{}

    stmt.QueryRow("W001").Scan(&result2.name, &result2.grade)

    fmt.Printf("name: %s\ngrade: %d\n", result2.name, result2.grade)

    var result3 = student{}

    stmt.QueryRow("B001").Scan(&result3.name, &result3.grade)

    fmt.Printf("name: %s\ngrade: %d\n", result3.name, result3.grade)

}

func main() {

    sqlPrepare()

}

Akan menampilkan output



**56.6 Insert,Update & Delete Data Menggunakan**

**Exec()**

package main

import (

    "database/sql"

    "fmt"

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

)

type student struct {

    id    string

    name  string

    age   int

    grade int

}

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

    db, err := sql.Open("mysql", "root:@tcp(127.0.0.1:3306)/farah\_belajar\_golang")

    if err != nil {

        return nil, err

    }

    return db, nil

}

func sqlExec() {

    db, err := connect()

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    defer db.Close()

    \_, err = db.Exec("insert into tb\_student values (?, ?, ?, ?)", "G001", "Galahad", 29, 2)

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    fmt.Println("insert success!")

    \_, err = db.Exec("update tb\_student set age = ? where id = ?", 28, "G001")

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    fmt.Println("update success!")

    \_, err = db.Exec("delete from tb\_student where id = ?", "G001")

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    fmt.Println("delete success!")

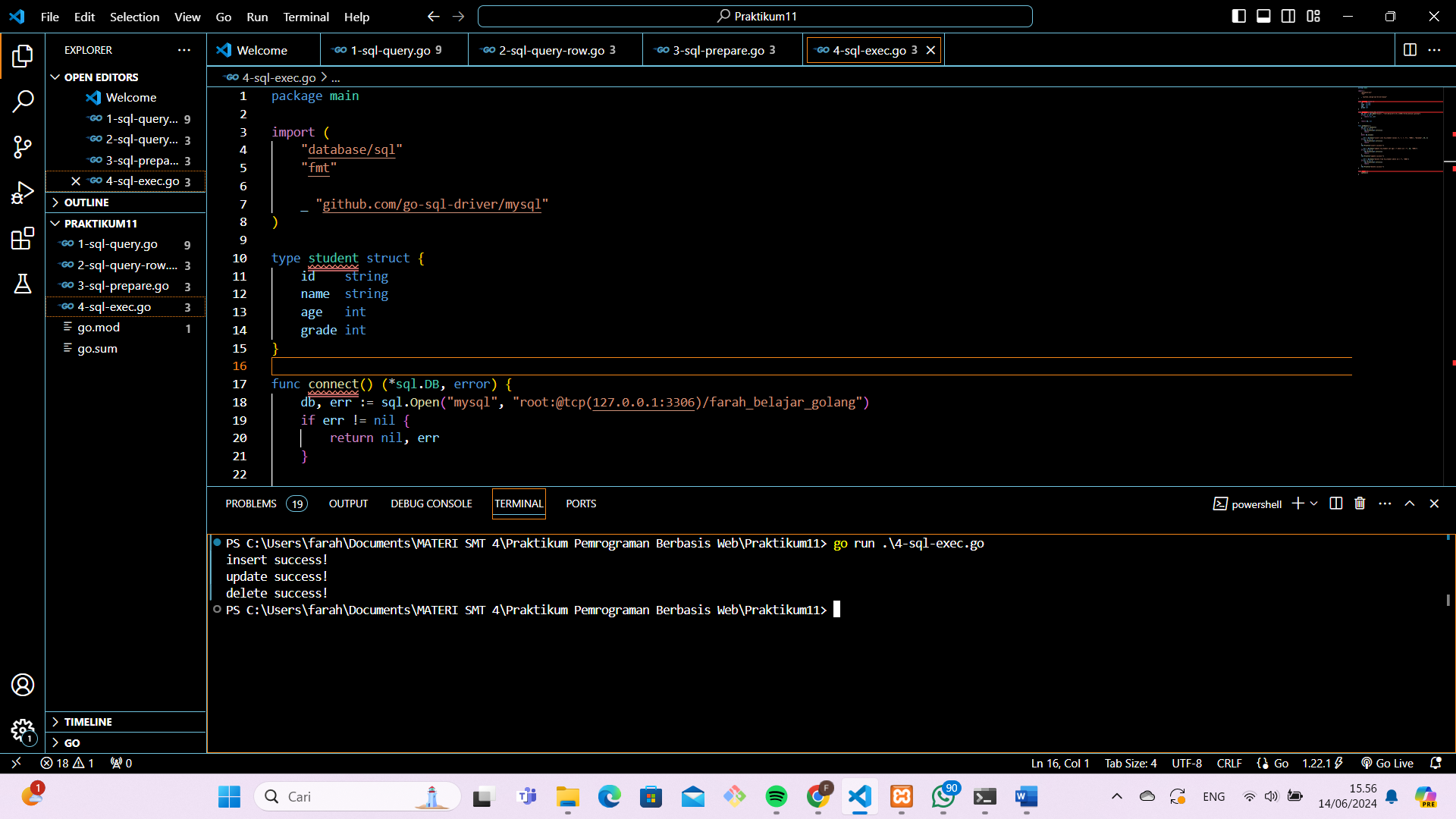
}

func main() {

    sqlExec()

}

Akan menampilkan output



package main

import (

    "database/sql"

    "fmt"

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

)

type student struct {

    id    string

    name  string

    age   int

    grade int

}

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

    db, err := sql.Open("mysql", "root:@tcp(127.0.0.1:3306)/farah\_belajar\_golang")

    if err != nil {

        return nil, err

    }

    return db, nil

}

func sqlExec() {

    db, err := connect()

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    defer db.Close()

    stmt, err := db.Prepare("insert into tb\_student values (?, ?, ?, ?)")

    stmt.Exec("G001", "Galahad", 29, 2)

    if err != nil {

        fmt.Println(err.Error())

        return

    }

    fmt.Println("insert success!")

}

func main() {

    sqlExec()

}

Akan menampilkan output

