

ACTIVITY GOLANG PERTEMUAN 7

NAMA : HABBILLAH BASHUDA

NPM : 51422723

KELAS : 3IA08

MATERI : DATABASE

SOAL

1. Jelaskan mengapa pentingnya sebuah program terintegrasi dengan database ! (5 point)

Jawab :

Pentingnya sebuah program terintegrasi dengan database dapat dijelaskan dari berbagai perspektif, baik dari sisi fungsionalitas, efisiensi, maupun pengelolaan data. Karena dengan itu penyimpanan dan pengelolaan data akan lebih efisien, akses data cepat dan tepat, keamanan data dan lebih terstruktur.

2. Dalam konteks pengembangan REST API dengan Golang, jelaskan mengapa pemilihan struktur project (project structure) yang tepat itu penting dan sebutkan minimal 3 folder/package yang umumnya ada dalam struktur project Golang! (5 point)

Jawab :

• Pemeliharaan yang Mudah:

- Struktur proyek yang baik memudahkan pengembang untuk memahami, mengembangkan, dan memelihara kode. Dengan struktur yang jelas, pengembang baru yang bergabung dengan tim bisa lebih cepat memahami alur dan komponen aplikasi tanpa merasa bingung mencari file atau kode tertentu.
- Memiliki folder-folder yang terorganisir dengan baik memungkinkan pemeliharaan yang lebih mudah dalam jangka panjang. Misalnya, memisahkan kode yang menangani logika bisnis, database, dan API endpoints akan mempercepat troubleshooting dan perbaikan bug.

• Kolaborasi yang Efektif:

- Dalam pengembangan tim, struktur proyek yang baik memungkinkan anggota tim untuk bekerja pada bagian yang berbeda dari aplikasi tanpa mengganggu satu sama lain. Sebagai contoh, satu anggota tim bisa bekerja pada bagian API handler sementara yang lain bekerja pada model data atau logika aplikasi.
- Struktur yang rapi juga membantu mengurangi konflik penggabungan (merge conflicts) dalam kontrol versi (seperti Git), karena kode lebih tersegmentasi.

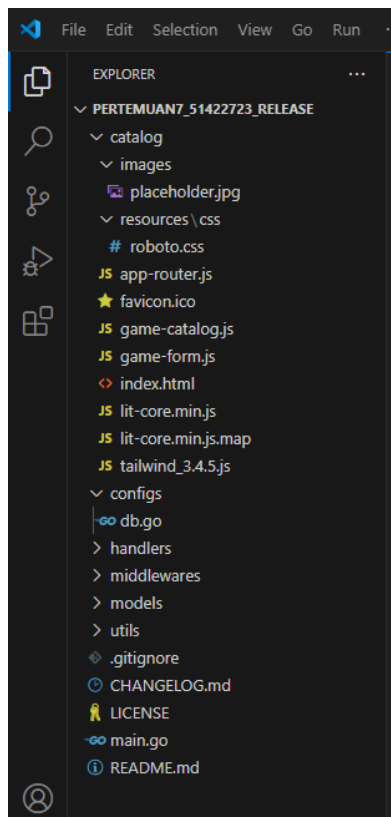
- **Skalabilitas dan Ekstensi:**

- Seiring berkembangnya aplikasi, struktur proyek yang baik memungkinkan penambahan fitur baru atau refactor kode dengan lebih mudah tanpa merusak bagian lain dari aplikasi.
- Memisahkan kode menjadi komponen-komponen yang lebih kecil dan terorganisir memungkinkan pengembangan aplikasi yang lebih skalabel dan mudah dikembangkan ke depannya.

- **Pemahaman yang Lebih Cepat untuk Pengembang Baru:**

- Struktur proyek yang konsisten dan terorganisir membantu pengembang baru untuk langsung memahami bagaimana aplikasi tersebut berfungsi dan bagaimana mereka bisa berkontribusi dalam pengembangan lebih lanjut.

Jawaban ACT



1.

```

Query OK, 0 rows affected (0.009 sec)

MariaDB [steam_catalog]> desc games;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_game | int(11) | NO | PRI | NULL | auto_increment |
| nama_game | varchar(100) | YES | | NULL | |
| developer | varchar(100) | YES | | NULL | |
| genre | varchar(100) | YES | | NULL | |
| harga | float | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.015 sec)

```

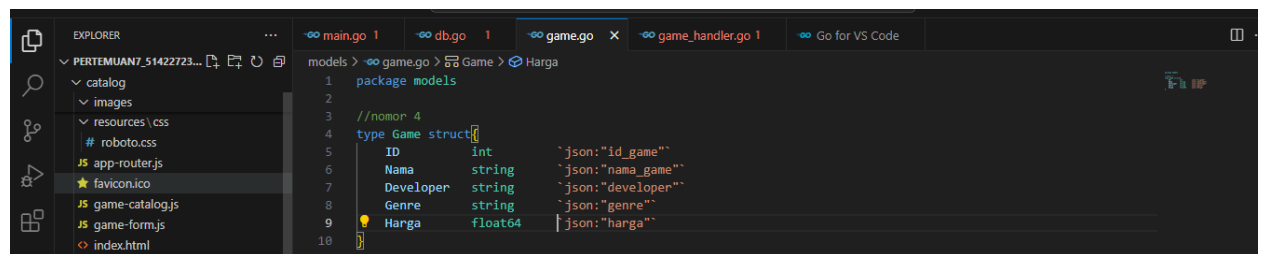
2. MariaDB [steam_catalog]>

```

17
18 //nomor2
19 configs.ConnectDB()
20 if configs.DB == nil{
21     log.Fatal("Database connection failed :")
22 }
23
24 defer func() {
25     if err := configs.DB.Close(); err != nil{
26         log.Printf("Error closing database connection: %v", err)
27     }
28 }()
29
30

```

3.



4.

5.

```

utils > utils.go > RespondJSON
1  package utils
2
3  import (
4      "encoding/json"
5      "net/http"
6  )
7
8  //nomor 5
9  func RespondJSON(w http.ResponseWriter, data interface{}) {
10     w.Header().Set("Content-Type", "application/json")
11     json.NewEncoder(w).Encode(data)
12 }
  
```

6.

```

handlers > serve_static.go > ServeStaticFile
1  package handlers
2
3  import (
4      "net/http"
5      "os"
6      "path/filepath"
7      "strings"
8  )
9
10 //nomor 6
11 func ServeStaticFile(w http.ResponseWriter, r *http.Request, baseDir string, fileServer http.Handler) {
12     path := strings.TrimPrefix(r.URL.Path, "/")
13     fullpath := filepath.Join(baseDir, path)
14
15     if strings.HasPrefix(r.URL.Path, "/api/") {
16         http.NotFound(w, r)
17         return
18     }
19
20     _, err := os.Stat(fullpath)
21     if err != nil {
22         http.Redirect(w, r, "/", http.StatusTemporaryRedirect)
23         return
24     }
25     fileServer.ServeHTTP(w, r)
26 }
  
```

7.

```

package handlers

import (
    "database/sql"
    "encoding/json"
    "net/http"
    "pertemuan7_51422723_release/configs"
    "pertemuan7_51422723_release/models"
    "pertemuan7_51422723_release/utils"
    "strconv"
)

//nomor 7
func HandleGames(w http.ResponseWriter, r *http.Request) {
    path := r.URL.Path
    var id int
  
```

```

var err error

if len(path) > len("/api/games/") {
    id, err = strconv.Atoi(path[len("/api/games/"):])
    if err != nil {
        http.Error(w, "ID tidak valid", http.StatusBadRequest)
        return
    }
} else {
    id = 0
}

switch r.Method {
case http.MethodGet:
    handleGetGames(w, id)
case http.MethodPost:
    handlePostGame(w, r)
case http.MethodPut:
    handleUpdateGame(w, r, id)
case http.MethodDelete:
    handleDeleteGame(w, id)
}
}

func handleGetGames(w http.ResponseWriter, id int) {
    if id == 0 {
        rows, err := configs.DB.Query("SELECT id_game, nama_game, developer,
genre, harga FROM games")
        if err != nil {
            http.Error(w, "Gagal mengambil data game",
http.StatusInternalServerError)
            return
        }
        defer rows.Close()

        var games []models.Game
        for rows.Next() {
            var game models.Game
            err := rows.Scan(&game.ID, &game>Nama, &game.Developer, &game.Genre,
&game.Harga)
            if err != nil {
                http.Error(w, "Gagal memproses data game",
http.StatusInternalServerError)
                return
            }
        }
    }
}

```

```

        games = append(games, game)
    }
    utils.RespondJSON(w, games)
} else {
    var game models.Game
    err := configs.DB.QueryRow("SELECT id_game, nama_game, developer, genre,
harga FROM games WHERE id_game = ?", id).
        Scan(&game.ID, &game>Nama, &game.Developer, &game.Genre, &game.Harga)
    if err != nil {
        if err == sql.ErrNoRows {
            http.Error(w, "Game tidak ditemukan", http.StatusNotFound)
        } else {
            http.Error(w, "Gagal mengambil data game",
http.StatusInternalServerError)
        }
        return
    }
    utils.RespondJSON(w, game)
}
}

func handlePostGame(w http.ResponseWriter, r *http.Request) {
    var game models.Game
    if err := json.NewDecoder(r.Body).Decode(&game); err != nil {
        utils.RespondJSON(w, map[string]string{"error": "Data tidak valid"})
        return
    }

    _, err := configs.DB.Exec(
        "INSERT INTO games (nama_game, developer, genre, harga) VALUES (?, ?, ?,
?)",
        game>Nama, game.Developer, game.Genre, game.Harga)

    if err != nil {
        // This will be caught by your LogRequestHandler middleware
        utils.RespondJSON(w, map[string]string{"error": err.Error()})
        return
    }

    utils.RespondJSON(w, map[string]string{"message": "Game berhasil
ditambahkan!"})
}

func handleUpdateGame(w http.ResponseWriter, r *http.Request, id int) {
    var game models.Game

```

```

    if err := json.NewDecoder(r.Body).Decode(&game); err != nil {
        http.Error(w, "Data tidak valid", http.StatusBadRequest)
        return
    }
    result, err := configs.DB.Exec("UPDATE games SET nama_game=?, developer=?,
genre=?, harga=? WHERE id_game=?",
        game.Nama, game.Developer, game.Genre, game.Harga, id)
    if err != nil {
        http.Error(w, "Gagal memperbarui game", http.StatusInternalServerError)
        return
    }
    rowsAffected, _ := result.RowsAffected()
    if rowsAffected == 0 {
        http.Error(w, "Game dengan ID tersebut tidak ditemukan",
http.StatusNotFound)
        return
    }
    utils.RespondJSON(w, map[string]string{"message": "Game berhasil
diperbarui!"})
}

func handleDeleteGame(w http.ResponseWriter, id int) {
    result, err := configs.DB.Exec("DELETE FROM games WHERE id_game=?", id)
    if err != nil {
        http.Error(w, "Gagal menghapus game", http.StatusInternalServerError)
        return
    }

    rowsAffected, _ := result.RowsAffected()
    if rowsAffected == 0 {
        http.Error(w, "Game dengan ID tersebut tidak ditemukan",
http.StatusNotFound)
        return
    }

    utils.RespondJSON(w, map[string]string{"message": "Game berhasil dihapus!"})
}

```

8.

The screenshot shows the VS Code interface with the Explorer on the left displaying a project structure for 'PERTEMUAN7_51422723_RELEASE'. The main editor shows the 'middlewares' directory with a file 'LogRequestHandler.go' open. The code defines a middleware function that logs incoming requests.

```
1 package middlewares
2
3 import (
4     "log"
5     "net/http"
6 )
7
8 func LogRequestHandler(next http.Handler) http.Handler {
9     //nomor 8
10    return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request){
11        log.Printf("incoming Request: %s %s", r.Method, r.URL.Path)
12        next.ServeHTTP(w,r)
13    })
14 }
15
16
```

9.

The screenshot shows the VS Code interface with the main.go file open. The code sets up a web server with a database connection and a file server. The terminal at the bottom shows the output of the application, including incoming requests and database connection logs.

```
11 _ "github.com/go-sql-driver/mysql"
12 )
13
14 func main() {
15     PORT := 22723 // Change the port to the last 5 digits of your student ID (NPM).
16
17     //nomor2
18     configs.ConnectDB()
19     if configs.DB == nil {
20         log.Fatal("Database connection failed :")
21     }
22
23     defer func() {
24         if err := configs.DB.Close(); err != nil {
25             log.Printf("Error closing database connection: %v", err)
26         }
27     }()
28
29     mux := http.NewServeMux()
30     fileServer := http.FileServer(http.Dir("catalog"))
```

Terminal Output:

```
2024/11/30 09:20:23 incoming Request: GET /
2024/11/30 09:20:23 incoming Request: GET /game-catalog.js
2024/11/30 09:20:23 incoming Request: GET /lit-core.min.js
2024/11/30 09:20:23 incoming Request: GET /game-form.js
2024/11/30 09:20:23 incoming Request: GET /app-router.js
2024/11/30 09:20:23 incoming Request: GET /resources/css/roboto.css
2024/11/30 09:20:23 incoming Request: GET /api/games
2024/11/30 09:20:23 incoming Request: GET /images/placeholder.jpg
2024/11/30 09:20:23 incoming Request: GET /favicon.ico
2024/11/30 09:20:54 incoming Request: POST /api/games
2024/11/30 09:20:54 incoming Request: GET /api/games
```


Add New Game

Nama Game

Developer

Genre

Harga

Add Game

Game Catalog

Add Game



dfdf

Developer: dfdf

Genre: dfdfdf

Price: 1000

Edit

Delete



fast and furious

Developer: 12345

Genre: action

Price: 1111

Edit

Delete



valorant

Developer: landfoqejb

Genre: wfnkbwikb

Price: 0

Edit

Delete



mobile legends

Developer: jenvb u

Genre: kvbivhb

Price: 0


Edit

Delete

Edit

Game Catalog

Add Game




gartic io

Developer: dfdfdf

Genre: dfdfdf

Price: 1000

EditDelete




fast and furious

Developer: 12345

Genre: action

Price: 1111

EditDelete




valorant

Developer: landfoqejb

Genre: wfnkbwikb

Price: 0

EditDelete



mobile legends

Developer: jenvb u

Genre: kvbivhb

Price: 0

EditDelete