

GOLANG FOR INTERMEDIATE
PERTEMUAN 6
(Integrasi Lit Component dengan Golang)

NAMA : ADITYA RIZKY NUGROHO

NPM : 50422097

KELAS : 3IA08

1. Jelaskan apa yang Anda ketahui mengenai LitElement! (5 point)

LitElement merupakan sebuah library web components yang modern dan efisien. Library ini menggunakan bahasa pemrograman JavaScript versi terbaru yang memungkinkan pengembang dalam membuat custom elements dengan cara yang lebih mudah dan lebih cepat serta performa yang tinggi. LitElement merupakan penerus dari library polymer yang pertama kali dikembangkan untuk pembuatan web components. LitElement dibangun dengan menyempurnakan kekurangan – kekurangan yang dihadapi saat pengembangan polymer. Dikarenakan lit merupakan penerus dari polymer, keduanya memiliki kesamaan. Kedua library ini dapat membuat components bekerja seperti layaknya elemen bawaan dari HTML dan fitur yang declarative pada template HTML

2. Apa keunggulan menggunakan Web Component (seperti Polymer, LitElement, dan sebagainya) dibandingkan menggunakan framework besar seperti ReactJS, Vue, dan sebagainya)? (5 point)

ACT (Total 90 Point)

Sebuah perusahaan bernama "Menyala Abangku Corp." sedang mengembangkan sistem autentikasi berbasis web components menggunakan Lit. Anda ditugaskan untuk membuat sebuah aplikasi web sederhana yang menampilkan form login dan pesan rahasia menggunakan Web Components dengan Lit library. Aplikasi ini harus memiliki sistem autentikasi dasar dan dapat menampilkan pesan rahasia hanya untuk pengguna yang sudah login.

Lakukan langkah – langkah pengerjaan berikut ini

1. Import Library yang diperlukan! (5 point)

```
1 //no 1
2 package main
3
4 import (
5     "encoding/json"
6     "fmt"
7     "log"
8     "net/http"
9     "os"
10    "path/filepath"
11    "strings"
12 )
```

2. Deklarasikan variabel konstanta PORT dengan nilai sesuai lima digit terakhir NPM! (5 point)

```
//no 2
const PORT = 22097
```

3. Buatlah sebuah function untuk serve satu folder static! (10 point)

```
36
37 //no 3
38 func serveStaticFile(w http.ResponseWriter, r *http.Request, baseDir string, fileServer http.Handler) {
39     path := strings.TrimPrefix(r.URL.Path, "/")
40     fullPath := filepath.Join(baseDir, path)
41
42     fileInfo, err := os.Stat(fullPath)
43     if err != nil {
44         if os.IsNotExist(err) {
45             log.Printf("File does not exist: %s", fullPath)
46         } else {
47             log.Printf("Error checking file: %s. Error: %v", fullPath, err)
48         }
49     } else {
50         log.Printf("retrieved %d bytes", fileInfo.Size())
51     }
52
53     fileServer.ServeHTTP(w, r)
54 }
```

4. Buatlah login handler untuk login api! (10 point)

```
// no 4
func handleLogin(w http.ResponseWriter, r *http.Request) {
    if r.Method != http.MethodPost {
        http.Error(w, "Method not allowed", http.
            StatusMethodNotAllowed)
        return
    }

    var loginRequest struct {
        Username string `json:"username"`
        Password string `json:"password"`
    }

    err := json.NewDecoder(r.Body).Decode(&loginRequest)
    if err != nil {
        http.Error(w, "Invalid request body", http.
            StatusBadRequest)
        return
    }

    if loginRequest.Username == "Aditya" && loginRequest.
        Password == "50422097" {
        w.WriteHeader(http.StatusOK)
        json.NewEncoder(w).Encode(map[string]string
            {"message": "Login successful"})
    } else {
        w.WriteHeader(http.StatusUnauthorized)
        json.NewEncoder(w).Encode(map[string]string
            {"message": "Invalid username or password"})
    }
}
}
```

5. Buat function main untuk menjalankan server, buatlah route API endpoint /api/login untuk fungsi login handler (handleLogin) (10 point).

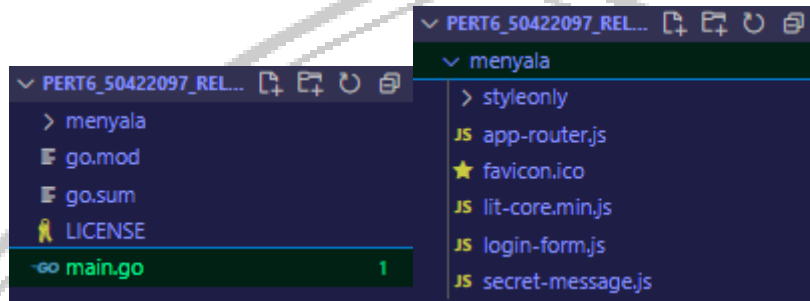
```
// no 5
func main() {
    fileServer := http.FileServer(http.Dir("menyala"))

    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        serveStaticFile(w, r, "menyala", fileServer)
    })

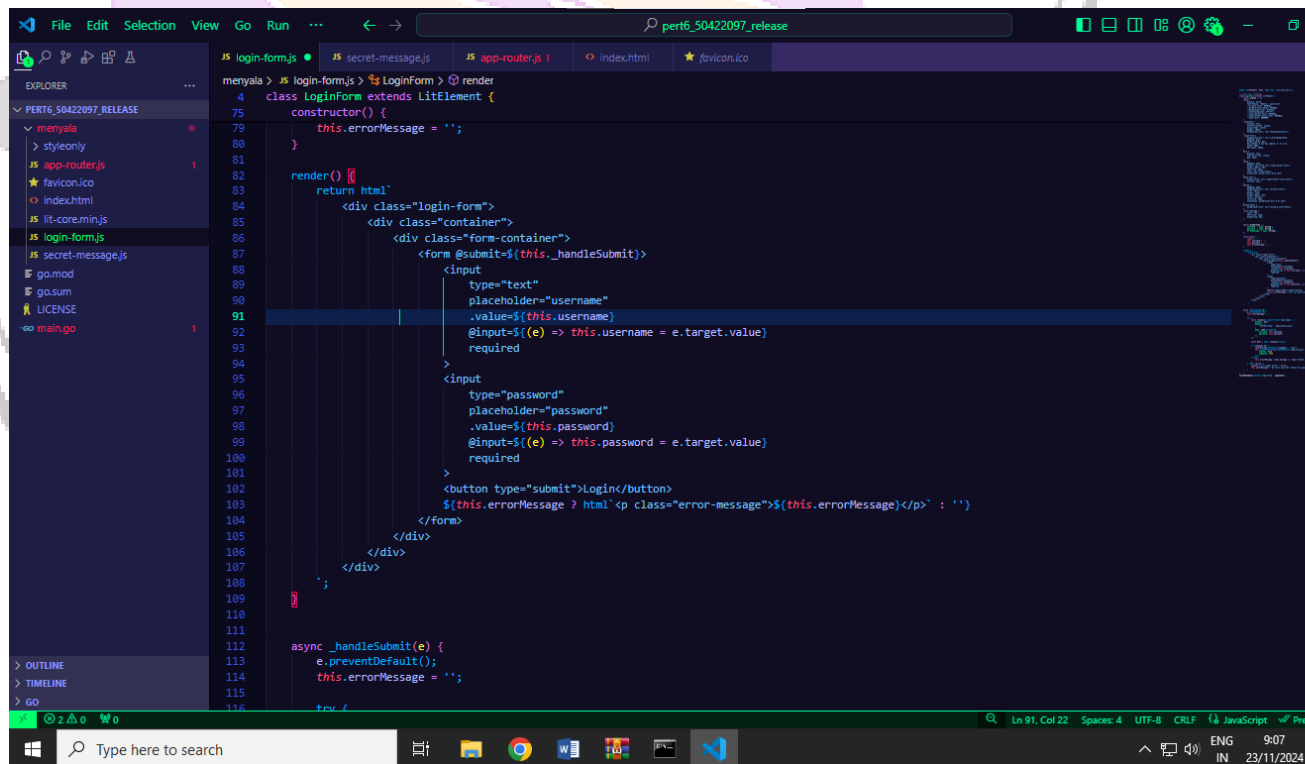
    http.HandleFunc("/api/login", handleLogin)

    log.Printf("Server running on http://[::1]:%d/ ...", PORT)
    err := http.ListenAndServe(fmt.Sprintf(":%d", PORT), nil)
    if err != nil {
        log.Fatalf("Error starting the server on port %d: %v", PORT, err)
    }
}
```

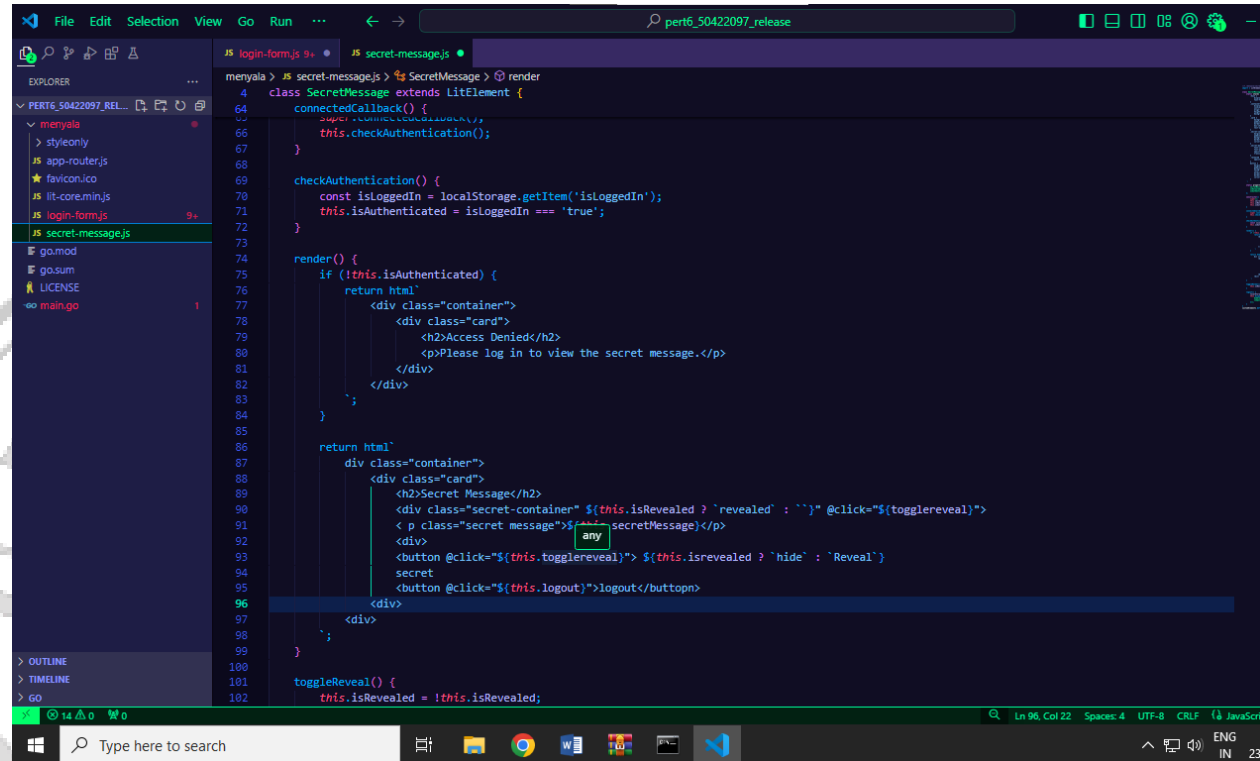
6. (5 point) Buat folder bernama `menyala` , kemudian masukkan file `lit-core.min.js` ke dalamnya, lalu buat empat file bernama:
- `index.html`
 - `app-router.js`
 - `login-form.js`
 - `secret-message.js`



7. Isikan Komponen `login-form.js` (15 point) dengan spesifikasi:
- Class Component (5 point)
 - Form Template (5 point)
 - Submit Handler (5 point)

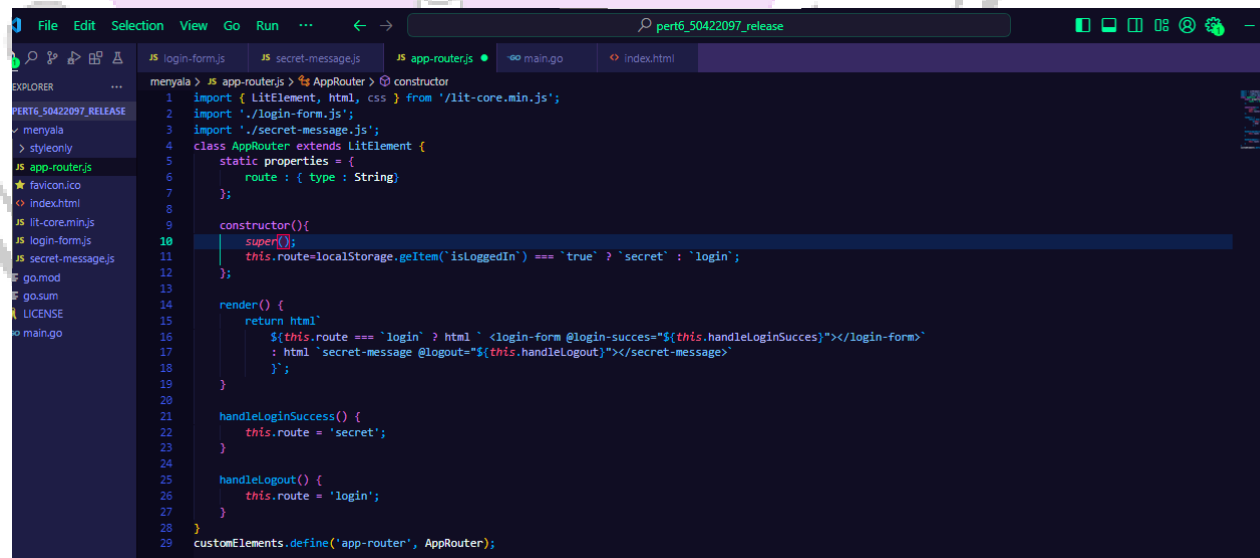


8. Buat file `secret-message.js` (15 point) yang menampilkan pesan rahasia untuk user yang sudah login.
- Implementasi Basic Component (5 point).
 - Implementasi Template dan User Interface (10 point).



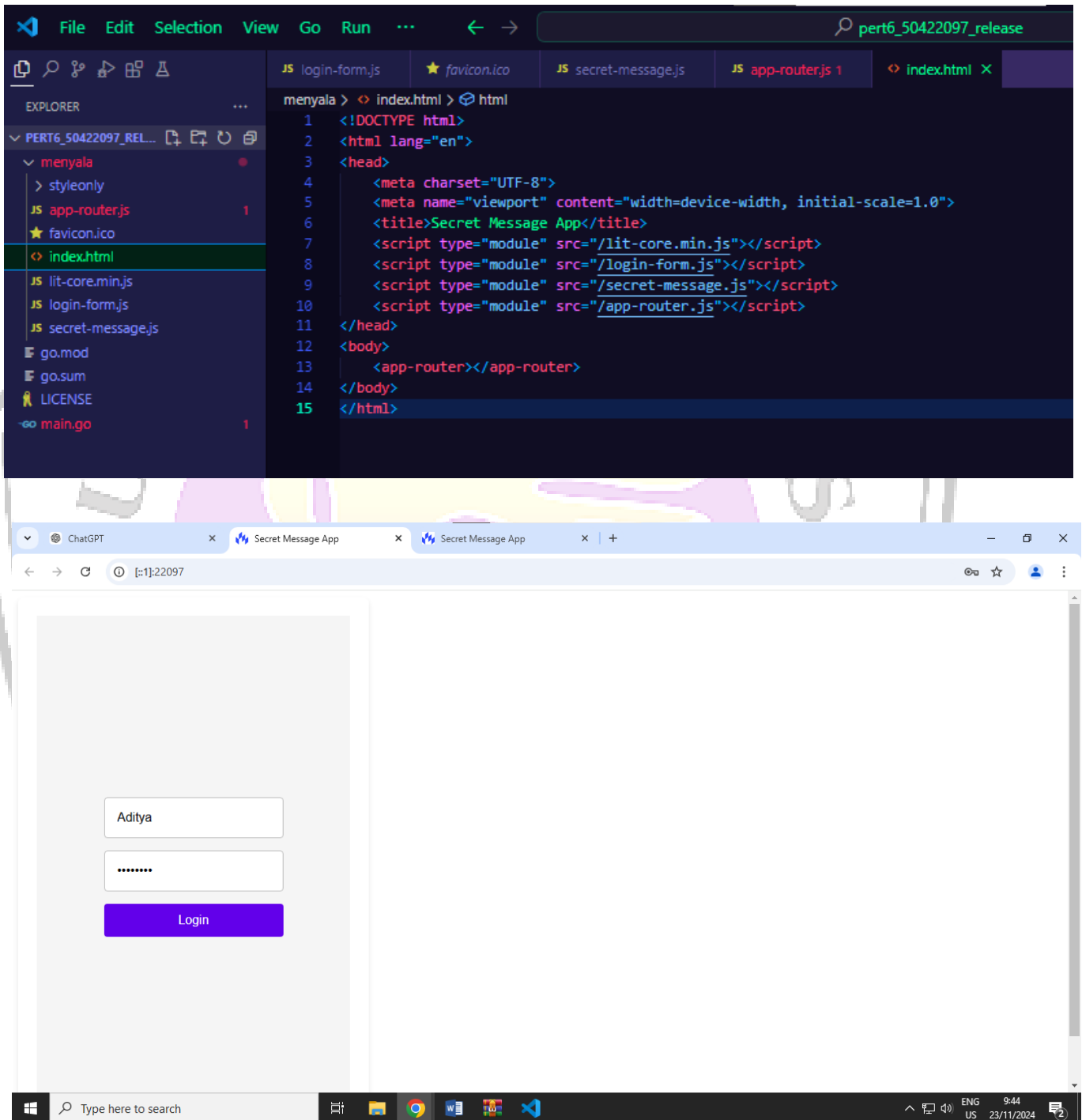
```
menyala > JS secret-message.js > SecretMessage > render
4 class SecretMessage extends LitElement {
5   connectedCallback() {
6     super.connectedCallback();
7     this.checkAuthentication();
8   }
9
10  checkAuthentication() {
11    const isLoggedIn = localStorage.getItem('isLoggedIn');
12    this.isAuthenticated = isLoggedIn === 'true';
13  }
14
15  render() {
16    if (!this.isAuthenticated) {
17      return html`
18        <div class="container">
19          <div class="card">
20            <h2>Access Denied</h2>
21            <p>Please log in to view the secret message.</p>
22          </div>
23        </div>
24      `;
25    }
26
27    return html`
28      <div class="container">
29        <div class="card">
30          <h2>Secret Message</h2>
31          <div class="secret-container" ${this.isRevealed ? 'revealed' : ''} @click=${this.toggleReveal}>
32            <p class="secret message">${secretMessage}</p>
33            <div>
34              <button @click=${this.toggleReveal}> ${this.isRevealed ? 'hide' : 'Reveal'}
35            </div>
36          </div>
37          <button @click=${this.logout}>logout</button>
38        </div>
39      </div>
40    `;
41  }
42
43  toggleReveal() {
44    this.isRevealed = !this.isRevealed;
45  }
46}
```

9. Isikan `app-router.js` (10 point) dengan spesifikasi:
- Implementasi Router Component:



```
menyala > JS app-router.js > AppRouter > constructor
1 import { LitElement, html, css } from './lit-core.min.js';
2 import './login-form.js';
3 import './secret-message.js';
4 class AppRouter extends LitElement {
5   static properties = {
6     route: { type: String };
7   };
8
9   constructor() {
10    super();
11    this.route = localStorage.getItem('isLoggedIn') === 'true' ? 'secret' : 'login';
12  }
13
14  render() {
15    return html`
16      ${this.route === 'login' ? html`<login-form @login-success=${this.handleLoginSuccess}></login-form>`
17      : html`secret-message @logout=${this.handleLogout}></secret-message>`}
18    `;
19  }
20
21  handleLoginSuccess() {
22    this.route = 'secret';
23  }
24
25  handleLogout() {
26    this.route = 'login';
27  }
28
29  customElements.define('app-router', AppRouter);
}
```

10. Buat Index.html (5 point) yang berfungsi sebagai entry point aplikasi web yang menghubungkan semua komponen web.



KENDALA

← → ↻ ⓘ [::1]:22097

```
secret-message @logout="handleLogout() { this.route = 'login'; }">
```

