

Tugas Web Design II



SIKLUS OBJECT VUE.JS

Nama : Nurhidayat

NIM : D112121057

Kelas : IF-4K

1. Step By Step

Berikut adalah langkah-langkah untuk membuat file HTML berisi script Vue.js dengan operasi CRUD menggunakan siklus objek untuk daftar makanan:

1. Siapkan editor teks atau IDE untuk menulis kode. Saya merekomendasikan menggunakan Visual Studio Code atau Sublime Text.
2. Buat file HTML kosong dan simpan dengan nama yang sesuai, misalnya index.html.
3. Masukkan kode HTML dasar ke dalam file dan tambahkan elemen <div> dengan id yang akan digunakan sebagai kontainer aplikasi Vue.js:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Food List</title>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  </head>
  <body>
    <div id="app">
      <!-- konten Vue.js akan ditambahkan di sini -->
    </div>
    <script>
      new Vue({
        el: "#app",
        // data dan methods akan ditambahkan di sini
      });
    </script>
  </body>
</html>
```

4. Tambahkan properti data ke dalam objek Vue dan tentukan array items sebagai daftar makanan awal:

```
<script>
  new Vue({
    el: '#app',
    data: {
      foods: ['Baso Urat', 'Teh Manis', 'Mie Ayam'],
      newFood: '',
      editedFood: null
    },
  },
```

5. Tambahkan kode HTML untuk menampilkan daftar makanan menggunakan v-for:

```
<ul>
  <li v-for="(food, index) in foods" :key="index">
    {{ food }}
    <button @click="editFood(index)">Edit</button>
    <button @click="deleteFood(index)">Hapus</button>
  </li>
</ul>
```

6. Tambahkan formulir untuk menambahkan item baru ke daftar:

```
<input type="text" v-model="newFood">
<button @click="editedFood !== null ? updateFood() : addFood()">{{ editedFood !== null ?
'Update' : 'Simpan' }}</button>
<button v-if="editedFood !== null" type="button" @click="cancelEdit()">Batal</button>
```

7. Tambahkan metode addItem ke dalam objek Vue untuk menambahkan item baru ke dalam daftar:

```
methods: {
  addFood: function() {
    if (this.newFood.trim() !== '') {
      this.foods.push(this.newFood.trim());
      this.newFood = '';
    }
  },
},
```

8. Tambahkan metode `addItem` ke dalam objek `Vue` untuk menambahkan item baru ke dalam daftar:

```
deleteFood: function(index) {  
  |   this.foods.splice(index, 1);  
  | }  
}
```

9. Tambahkan tombol "Edit" ke setiap item dalam daftar dan tambahkan metode `editItem` untuk membuka mode pengeditan:

```
editFood: function(index) {  
  |   this.editedFood = index;  
  |   this.newFood = this.foods[index];  
  |   console.log(this.editFood);  
  | },  
  | updateFood: function() {
```

10. Tambahkan formulir untuk memperbarui nilai item yang dipilih dan tambahkan metode `saveItem` dan `cancelEdit` untuk menyimpan atau membatalkan perubahan:

```
editFood: function(index) {  
  |   this.editedFood = index;  
  |   this.newFood = this.foods[index];  
  |   console.log(this.editFood);  
  | },  
  | updateFood: function() {  
  |   | if (this.editedFood !== null) {  
  |   |   | if (this.newFood.trim() !== '') {  
  |   |   |   | this.foods.splice(this.editedFood, 1, this.newFood.trim());  
  |   |   |   | this.newFood = '';  
  |   |   |   | this.editedFood = null;  
  |   |   | }  
  |   | }  
  | },  
  | cancelEdit() {  
  |   | this.newFood = '';  
  |   | this.editedFood = null;  
  | },  
}
```

11. Lalu Tes pada console browser dengan menambahkan , mengedit atau menghapus item dari daftar makanan.

2. Penjelasan Code

1. `<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>`

Baris kode ini memuat Vue.js, yang adalah framework JavaScript yang digunakan untuk membuat aplikasi single-page. Vue.js memiliki banyak fitur, termasuk kemampuan untuk membuat komponen, memanipulasi DOM, mengelola state, dan banyak lagi. Dalam hal ini, saya menggunakan Vue.js untuk membuat aplikasi CRUD sederhana yang dapat menambahkan, mengedit, dan menghapus item dari daftar makanan.

2. `addFood: function() {`

```
methods: {  
  addFood: function() {  
    if (this.newFood.trim() !== '') {  
      this.foods.push(this.newFood.trim());  
      this.newFood = '';  
    }  
  },  
}
```

`addFood` adalah method Vue.js yang ditambahkan ke instance Vue dengan menggunakan opsi `methods`. Method ini akan dipanggil saat pengguna menambahkan makanan baru ke daftar.

Dalam method `addFood`, saya menambahkan nilai `newItem` (yaitu makanan baru yang dimasukkan oleh pengguna) ke array `items` menggunakan method `push()`. Kemudian, saya mengatur nilai `newItem` menjadi string kosong lagi, sehingga input makanan di formulir dikosongkan.

Method `addFood` kemudian dipanggil saat formulir disubmit, menggunakan atribut `v-on:submit.prevent`. Atribut ini menunjukkan bahwa saya ingin menghentikan perilaku default dari form submit, yang dalam hal ini adalah untuk memuat ulang halaman. Sebagai gantinya, saya ingin menambahkan makanan baru ke daftar tanpa memuat ulang halaman.

3. `editFood: function(index) {`

`editFood` adalah method Vue.js yang akan dipanggil saat pengguna mengklik tombol "Edit" pada item daftar makanan. Method ini menerima parameter `index`, yang adalah indeks item yang ingin di-edit.

Dalam method `editFood`, saya mengatur nilai `editingIndex` (yaitu indeks item yang sedang diedit) menjadi `index`, dan nilai `editingItem` (yaitu item yang sedang diedit) menjadi item di array `items` dengan indeks `index`. Dengan cara ini, saya dapat menampilkan formulir editing dengan input teks yang berisi nilai item yang sedang diedit, dan tombol "Simpan" dan "Batal".

Ketika tombol "Simpan" pada formulir editing ditekan, method `saveFood` akan dipanggil dan akan menyimpan perubahan ke item daftar makanan. Sebaliknya, jika tombol "Batal" ditekan, method `cancelEdit` akan dipanggil dan akan membatalkan proses editing dan menampilkan item dalam mode tampilan.

4. updateFood: function() {

```
updateFood: function() {  
  if (this.editedFood !== null) {  
    if (this.newFood.trim() !== '') {  
      this.foods.splice(this.editedFood, 1, this.newFood.trim());  
      this.newFood = '';  
      this.editedFood = null;  
    }  
  }  
},
```

updateFood adalah method Vue.js yang akan dipanggil saat pengguna menekan tombol "Simpan" pada formulir editing setelah mengedit item daftar makanan.

Dalam method updateFood, saya mengganti nilai item di array items yang sedang diedit dengan nilai editingItem. Kemudian, saya mengatur nilai editingIndex menjadi -1 (yaitu keluar dari mode editing) dan editingItem menjadi null (karena tidak ada item yang sedang diedit).

Setelah itu, formulir editing akan disembunyikan dan item yang telah diperbarui akan ditampilkan dalam mode tampilan. Penting untuk dicatat bahwa saya mengganti item di array dengan indeks yang sama dengan item yang sedang diedit, sehingga nilai item yang lama akan diganti dengan nilai item yang baru.

5. deleteFood: function(index) {

```
deleteFood: function(index) {  
  this.foods.splice(index, 1);  
}
```

deleteFood adalah method Vue.js yang akan dipanggil saat pengguna mengklik tombol "Hapus" pada item daftar makanan. Method ini menerima parameter index, yaitu indeks item yang ingin dihapus.

Dalam method deleteFood, saya menggunakan method splice() untuk menghapus item dari array items dengan indeks index. Setelah item dihapus, tampilan daftar makanan akan diperbarui secara otomatis karena saya menggunakan v-for untuk menampilkan daftar makanan, dan saya telah menghapus item dari array items.

Penting untuk dicatat bahwa method splice() mengubah array secara langsung (in-place), sehingga saya tidak perlu mengembalikan nilai dari method deleteFood.

6. new Vue({

```
new Vue({  
  el: '#app',  
  data: {  
    foods: ['Baso Urat', 'Teh Manis', 'Mie Ayam '],  
    newFood: '',  
    editedFood: null  
  },  
},
```

`new Vue({})` adalah sintaks yang digunakan untuk membuat instance Vue.js yang baru. Pada sintaks tersebut, kita dapat menambahkan berbagai opsi konfigurasi Vue, seperti data, methods, computed, watch, dan lainnya. Dalam kasus kode yang telah diberikan, kita menambahkan beberapa opsi ke instance Vue, yaitu:

- `el`: Opsi ini menentukan elemen HTML tempat Vue.js diikatkan. Dalam kasus ini, elemen dengan ID `app` dipilih sebagai tempat Vue.js diikatkan.
- `data`: Opsi ini menentukan data yang akan digunakan oleh Vue.js. Dalam kasus ini, kita hanya memiliki satu data, yaitu array items yang berisi daftar makanan.
- `methods`: Opsi ini menentukan method Vue.js yang akan digunakan. Dalam kasus ini, kita memiliki beberapa method seperti `addFood`, `editFood`, `updateFood`, dan `deleteFood`.

Setelah membuat instance Vue, kita mengikatnya ke elemen HTML dengan ID `app` menggunakan sintaks `vm.$mount('#app')`.