

Nama : Yazid Dhiaulhaq Ismail

Kelas : 3DB03

NPM : 31120173

## UJIAN GOLANG INTERMEDIATE

### 1. Apa yang dimaksud dengan crud? (10 poin)

CRUD adalah singkatan dari Create, Read, Update, dan Delete. Ini adalah empat operasi dasar yang digunakan dalam pengelolaan data dalam sistem database atau aplikasi berbasis data. Setiap operasi memiliki makna berikut:

- **Create (Membuat):** Operasi ini digunakan untuk membuat entitas data baru dalam database. Misalnya, dalam aplikasi toko online, operasi Create dapat digunakan untuk membuat entri baru untuk produk yang akan dijual.
- **Read (Membaca):** Operasi ini digunakan untuk membaca atau mengambil data yang ada dari database. Misalnya, dalam aplikasi toko online, operasi Read dapat digunakan untuk menampilkan daftar produk yang tersedia kepada pengguna.
- **Update (Memperbarui):** Operasi ini digunakan untuk memperbarui data yang sudah ada dalam database. Misalnya, dalam aplikasi toko online, operasi Update dapat digunakan untuk mengubah harga atau deskripsi produk yang sudah ada.
- **Delete (Menghapus):** Operasi ini digunakan untuk menghapus data yang tidak diperlukan lagi dari database. Misalnya, dalam aplikasi toko online, operasi Delete dapat digunakan untuk menghapus produk yang tidak lagi tersedia atau tidak diperlukan.

### 2. Sebutkan struktur crud pada go Language! (10 poin)

Dalam bahasa Go, struktur CRUD (Create, Read, Update, Delete) biasanya diimplementasikan menggunakan operasi-operasi dasar dalam paket database/sql. Berikut adalah contoh struktur CRUD dalam Go Language:

#### a. Create (Membuat Data):

- Membuka koneksi ke database menggunakan `sql.Open`.
- Membuat kueri INSERT baru dengan `db.Prepare` dan `db.Exec`.
- Menangani kesalahan yang terjadi saat menyimpan data baru ke database.

#### b. Read (Membaca Data):

- Membuka koneksi ke database menggunakan `sql.Open`.

- Membuat kueri SELECT dengan db.Query atau db.QueryRow.
- Menggunakan Scan untuk membaca hasil query dan mengisi data ke struct atau variabel.
- Menangani kesalahan yang terjadi saat membaca data dari database.

c. Update (Mengubah Data):

- Membuka koneksi ke database menggunakan sql.Open.
- Membuat kueri UPDATE dengan db.Prepare dan db.Exec.
- Menangani kesalahan yang terjadi saat mengubah data di database.

d. Delete (Menghapus Data):

- Membuka koneksi ke database menggunakan sql.Open.
- Membuat kueri DELETE dengan db.Prepare dan db.Exec.
- Menangani kesalahan yang terjadi saat menghapus data dari database

3. Jelaskan fungsi dan tujuan go build serta go run ! (10 poin)

- Go build: Perintah ini digunakan untuk mengkompilasi paket dan dependensinya.
- Go run: Perintah ini digunakan untuk mengkompilasi dan menjalankan program Go.

4. Membuat dan menginput data pada table database! (20 poin)

id	npm	Nama	Kelas	Profile
1	51416277	Azman Agung Nugraha	4IA17	gambar1.jpg
2	15117348	Rizky Imam Nugroho	3KA01	gambar1.jpg
3	52417067	Fahmi Ardhiansyah	3IA03	gambar1.jpg



✓ Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)

```
SELECT * FROM `mahasiswa`
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by

Extra options

				id	npm	nama	kelas	profile
<input type="checkbox"/>	Edit	Copy	Delete	8	51416277	Azman Agung Nugraha	4IA17	gambar1.jpg
<input type="checkbox"/>	Edit	Copy	Delete	9	15117348	Rizky Imam Nugroho	3KA01	gambar1.jpg
<input type="checkbox"/>	Edit	Copy	Delete	10	52417067	Fahmi Ardhiansyah	3IA03	gambar1.jpg

☐ Check all
 With selected:
 Edit
 Copy
 Delete
 Export

5. Buatlah output seperti gambar dibawah! (50 poin)

My App

Menu

Mahasiswa

### Data Mahasiswa

Data Mahasiswa yang terdapat pada database

TAMBAH DATA

Search..

#	Profile	Npm	Nama	Kelas	Action
1	u.jpg	51416277	Azman Agung Nugraha	4IA17	EDIT DELETE
2	u.jpg	15117348	Rizky Imam Nugroho	3KA01	EDIT DELETE
3	u.jpg	52417067	Fahmi Ardhiansyah	3IA03	EDIT DELETE

main.go

```
51 // Yazid Dhiaulhaq Ismail 31120173
52 func dbConn() (db *sql.DB) {
53     dbDriver := "mysql"
54     dbName := "yazid_31120173_ujian"
55     dbUser := "root"
56     dbPass := ""
57     db, err := sql.Open(dbDriver, dbUser+"."+dbPass+"@tcp(localhost)/"+dbName)
58
59     if err != nil {
60         panic(err.Error())
61     }
62
63     return db
64 }
```

my-app.html

```
72 <app-route route="{{route}}" pattern="{{rootPath}}:page" data="{{routeData}}" tail="{{subroute}}">
73 </app-route>
74
75 <app-drawer-layout fullbleed narrow="{{narrow}}">
76     <!-- Drawer content -->
77     <app-drawer id="drawer" slot="drawer" swipe-open="{{narrow}}">
78         <app-toolbar>Menu</app-toolbar>
79         <iron-selector selected="{{page}}" attr-for-selected="name" class="drawer-list" role="navigation">
80             <a name="mahasiswa" href="{{rootPath}}mahasiswa">Mahasiswa</a>
81             <!-- <a name="view2" href="{{rootPath}}view2">View Two</a>
82             <a name="view3" href="{{rootPath}}view3">View Three</a> -->
83         </iron-selector>
84     </app-drawer>
85
86     <!-- Main content -->
87     <app-header-layout has-scrolling-region>
88
89         <app-header slot="header" condenses reveals effects="waterfall">
90             <app-toolbar>
91                 <paper-icon-button icon="my-icons:menu" drawer-toggle></paper-icon-button>
92                 <div main-title>My App</div>
93             </app-toolbar>
94         </app-header>
95
96         <iron-pages selected="{{page}}" attr-for-selected="name" fallback-selection="view404" role="main">
97             <mahasiswa-page name="mahasiswa"></mahasiswa-page>
98             <!-- <my-view2 name="view2"></my-view2>
99             <my-view3 name="view3"></my-view3> -->
100             <my-view404 name="view404"></my-view404>
101         </iron-pages>
102     </app-header-layout>
```

```
107 Polymer.setPassiveTouchGestures(true);|
108 class MyApp extends Polymer.Element {
109   static get is() { return 'my-app'; }
110   static get properties() {
111     return {
112       page: {
113         type: String,
114         reflectToAttribute: true,
115         observer: '_pageChanged',
116       },
117       routeData: Object,
118       subroute: Object,
119       rootPath: String,
120     };
121   }
122   static get observers() {
123     return [
124       '_routePageChanged(routeData.page)',
125     ];
126   }
127   _routePageChanged(page) {
128     this.page = page || 'mahasiswa';
129     if (!this.$.drawer.persistent) {
130       this.$.drawer.close();
131     }
132   }
133   _pageChanged(page) {
134     const resolvedPageUrl = this.resolveUrl(page + '.html');
135     Polymer.importHref(
136       resolvedPageUrl,
137       null,
138       this.showPage404.bind(this),
```

mahasiswa.html

```
42 <div class="card">
43   <h1>Data Mahasiswa</h1>
44   <p>Data Mahasiswa yang terdapat pada database</p>
45   <template is="dom-if" if="[[success]]">
46     <p class="alert-success"><strong>Message:</strong> [[success]]</p>
47   </template>
48
49   <!-- Get data from rest api -->
50   <iron-ajax auto url="{{url}}" method="{{method}}" handle-as="json" content-type="application-json"
51   | last-response="{{response}}"></iron-ajax>
52   <paper-button raised class="link" style="margin-bottom: 20px;" on-click="openAddModal">Tambah Data</
53   paper-button>
54   <paper-input-container no-label-float>
55     <label slot="label">Search..</label>
56     <iron-input slot="input">
57       <input type="text" value="{{keyword::input}}" width="30%">
58     </iron-input>
59   </paper-input-container>
60
61   <!-- Datatable -->
62   <vaadin-grid items="{{response.data}}">
63     <vaadin-grid-column width="50px">
64       <template class="header">#</template>
65       <template>
66         <div>[[index]]</div>
67       </template>
68     </vaadin-grid-column>
69     <vaadin-grid-column width="80px">
70       <template class="header">Profile</template>
71       <template>
72         
73
74         
75       </template>
76     </vaadin-grid-column>
77     <vaadin-grid-column>
78       <template class="header">Npm</template>
79       <template>
80         <div>[[item.npm]]</div>
81       </template>
82     </vaadin-grid-column>
83     <vaadin-grid-column width="150px">
84       <template class="header">Nama</template>
85       <template>
86         <div>[[item.nama]]</div>
87       </template>
88     </vaadin-grid-column>
89     <vaadin-grid-column>
90       <template class="header">Kelas</template>
91       <template>
92         <div>[[item.kelas]]</div>
93       </template>
94     </vaadin-grid-column>
95     <vaadin-grid-column width="200px">
96       <template class="header">Action</template>
97       <template>
98         <paper-button id="[[item.id]]" class="warning" on-click="openEditModal">Edit
99       </paper-button>
100        <paper-button id="[[item.id]]" class="danger" on-click="openDeleteModal">Delete
101      </paper-button>
102    </template>
103  </vaadin-grid-column>
```

```

120 class MahasiswaPage extends Polymer.Element {
121   static get is() { return 'mahasiswa-page'; }
122
123   static get properties() {
124     return {
125       response: {
126         type: Object,
127       },
128       keyword: {
129         type: String,
130         value: ''
131       },
132       url: {
133         computed: '_computeUrl(keyword)'
134       },
135       method: {
136         type: String,
137         value: "GET"
138       },
139       success: String,
140       uid: {
141         type: Number,
142         notify: true
143       }
144     }
145   }
146
147   _computeUrl(keyword) {
148     if (keyword) {
149       return ['/mahasiswa/search', keyword].join('/');
150     } else {

```

form-add.html

```

29 <h2>Tambah Data User</h2>
30 <paper-input-container>
31   <iron-input slot="input">
32     <input type="text" value="{{formData.npm::input}}" placeholder="Npm">
33   </iron-input>
34 </paper-input-container>
35 <paper-input-container>
36   <iron-input slot="input">
37     <input type="text" value="{{formData.nama::input}}" placeholder="Nama">
38   </iron-input>
39 </paper-input-container>
40 <paper-input-container>
41   <iron-input slot="input">
42     <input type="text" value="{{formData.kelas::input}}" placeholder="Kelas">
43   </iron-input>
44 </paper-input-container>
45 <div class="buttons">
46   <paper-button raised class="link" on-click="addMahasiswa" style="margin-top: 20px;"
47     dialog-confirm>Tambah
48     Data
49   </paper-button>
50 </div>
51 <iron-ajax auto id="mahasiswaAjax" url="{{url}}" method="{{method}}" handle-as="text"
52   content-type="application/json" on-response="_handleInsert">
53 </iron-ajax>
54 </template>

```

```

69         type: String,
70         value: '/api/mahasiswa'
71     },
72     method: {
73         type: String,
74         value: 'GET'
75     },
76     success: {
77         type: String,
78         notify: true
79     },
80     response: {
81         type: Object,
82         notify: true
83     }
84 }
85
86 addMahasiswa() {
87     this.url = '/api/mahasiswa';
88     this.method = 'POST';
89     this.success = 'Data mahasiswa berhasil ditambahkan';
90     this.$.mahasiswaAjax.params = this.formData;
91     this.formData = {};
92 }
93 _handleInsert(e) {
94     this.method = 'GET';
95     this.set('response', JSON.parse(e.detail.response));
96 }
97 }
98 window.customElements.define(FormAdd.is, FormAdd);
99 </script>

```

## form-edit.html

```

18 <h2>Ubah Data Mahasiswa</h2>
19 <input type="hidden" value="{{formData.id::input}}">
20 <paper-input-container>
21 <iron-input slot="input">
22 <input type="text" value="{{formData.npm::input}}" placeholder="Npm">
23 </iron-input>
24 </paper-input-container>
25 <paper-input-container>
26 <iron-input slot="input">
27 <input type="text" value="{{formData.nama::input}}" placeholder="Nama">
28 </iron-input>
29 </paper-input-container>
30 <paper-input-container>
31 <iron-input slot="input">
32 <input type="text" value="{{formData.kelas::input}}" placeholder="Kelas">
33 </iron-input>
34 </paper-input-container>
35
36 <div class="buttons">
37 <paper-button raised class="link" on-click="editMahasiswa" style="margin-top: 20px;" dialog-confirm
38 Data
39 </paper-button>
40 </div>
41
42 <iron-ajax auto id="getMahasiswaAjax" url="{{mhsurl}}" method="GET" handle-as="json"
43 content-type="application/json" on-response="_handleGet">
44 </iron-ajax>
45 <iron-ajax auto id="editMahasiswaAjax" url="{{url}}" method="{{method}}" handle-as="text"
46 content-type="application/json" on-response="_handleUpdate">
47 </iron-ajax>
48 </template>

```



```
69         notify: true
70     },
71     response: {
72         type: Object,
73         notify: true
74     },
75     uid: Number,
76 }
77 }
78
79 _computeUrl(uid) {
80     return ['/api/mahasiswa', uid].join('/');
81 }
82
83
84 editMahasiswa() {
85     this.url = '/api/mahasiswa';
86     this.method = 'PUT';
87     this.success = 'Data mahasiswa berhasil diubah';
88     this.$.editMahasiswaAjax.params = this.formData;
89     this.formData = {};
90 }
91
92 _handleUpdate(e) {
93     this.method = 'GET';
94     this.set('response', JSON.parse(e.detail.response));
95 }
96
97 _handleGet(e) {
98     this.set('formData', e.detail.response.data);
99 }
```

## form-delete.html

```
18         <paper-button class="link" dialog-dismiss>Batal</paper-button>
19         <paper-button class="danger" on-click="deleteMahasiswa" raised dialog-confirm>Hapus</paper-button>
20     </div>
21     <iron-ajax auto url="{{url}}" method="{{method}}" handle-as="json" content-type="application/json"
22         on-response="_handleDelete"></iron-ajax>
23 </template>
24 <script>
25     class FormDelete extends Polymer.Element {
26         static get is() { return 'form-delete'; }
27         static get properties() {
28             return {
29                 uid: Number,
30                 url: String,
31                 method: String,
32                 success: {
33                     type: String,
34                     notify: true
35                 },
36                 response: {
37                     type: Object,
38                     notify: true
39                 }
40             }
41         }
42         deleteMahasiswa() {
43             this.method = 'DELETE';
44             this.url = '/api/mahasiswa/' + this.uid;
45             this.success = 'Data mahasiswa berhasil dihapus';
46         }
47         _handleDelete(e) {
48             this.method = 'GET';
```