

**Laporan Web Application**  
**Darvin Casing Store (DCS)**



**Darvin**  
**1906300694**  
**Sistem Basis Data**

Fakultas Teknik  
Universitas Indonesia  
Depok  
2021

## Laporan Tugas Web Application berbasis CRUD dan Rest API

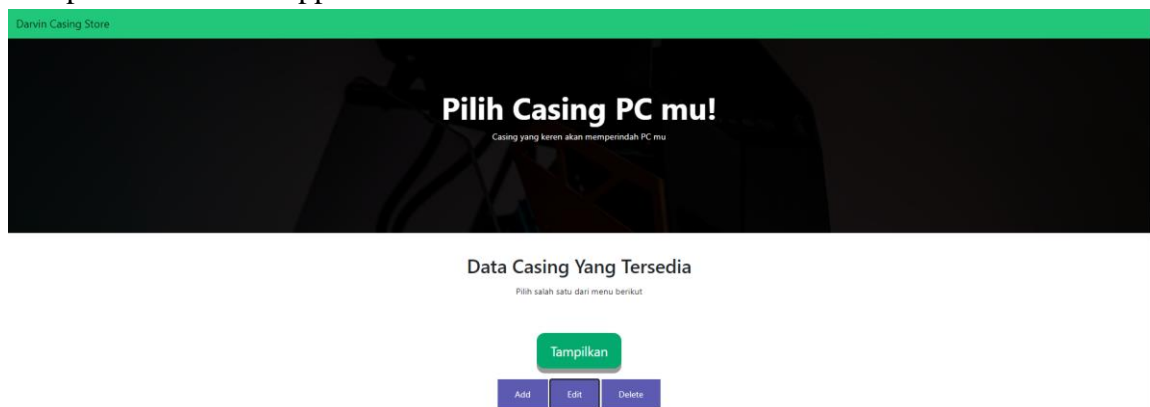
### Sistem Basis Data – 01

#### Deskripsi Aplikasi

Web application yang dibuat pada laporan ini merupakan sebuah website bernama DCS (Darvin Casing Store) dengan tujuan untuk mempermudah admin toko melakukan pengaturan beberapa casing yang dapat ditawarkan kepada pelanggan. Produk-produk casing yang ditampilkan juga memiliki beberapa deskripsi lainnya seperti form factor yang didukung, brand yang melakukan produksi, serta harga yang akan selalu di update ketika terdapat perubahan. Dalam website tersebut dapat dilakukan fungsi CRUD pada tabel casing yang tersedia yaitu operasi *Create* berupa penambahan produk pada tabel dalam website tersebut menggunakan metode POST, operasi *Read* yaitu untuk melakukan *fetching* dari database dan menampilkan produk dalam bentuk tabel kepada pelanggan menggunakan metode GET, operasi Update yaitu untuk mengatur setiap nilai atribut produk casing yang ada pada tabel dengan menggunakan metode PUT, dan operasi Delete untuk menghapus data casing ketika produk tersebut sudah *out of stock* atau tidak dijual lagi menggunakan metode DELETE.

Berikut merupakan penjelasan mengenai Web app yang telah dibuat berdasarkan fitur-fitur yang tersedia

#### 1. Tampilan awal Web App



Ketika membuka web app tersebut laman utama akan menunjukkan navigation bar dan juga tampilan awal dengan gambar casing pc. Selanjutnya terdapat beberapa menu yang dapat digunakan untuk pengaturan data produk Casing.

#### 2. Menampilkan tabel dengan menekan tombol Tampilkan untuk metode GET

Tampilkan

AddEditDelete

No	Nama	Form Factor	Brand	Price
1	Corsair 4000D Tempered Glass Black	ATX, MATX, ITX	Corsair	1379000
2	Lian Li LANCOOL 205 MESH BLACK	ATX, MATX, ITX	Lian Li	1080000
3	Corsair 4000X Tempered Glass White	ATX, MATX, ITX	Corsair	1850000
4	Corsair 465X RGB Tempered Glass Black	ATX, MATX, ITX	Corsair	2279000
5	NZXT H510 Elite White	ATX, MATX, ITX	NZXT	2450000
6	Metallic Gear Neo Qube	ATX, MATX, ITX	Metallic Gear	2000000
7	Cube Gaming Kellva White	MATX, ITX	Cube Gaming	410000
8	Corsair Carbide 275R Airflow Tempered Glass Black	ATX, MATX, ITX	Corsair	1229000
9	MSI MPG Sekira 100R	ATX, MATX, ITX	MSI	1194000
10	PRIME X-[S] Pink - Aluminium Tempered Glass	ATX, MATX, ITX	Prime	1200000

Ketika menekan tombol menampilkan, maka website akan menampilkan tabel berupa data Casing yang dapat terjadi karena melakukan metode POST pada node.js untuk diquery pada database. Selain itu, tombol tampilan ini juga digunakan untuk merefresh kembali table ketika terdapat perubahan seperti edit atau delete.

### 3. Menambahkan data baru pada tabel melalui tombol Add dengan metode POST

Tambah Casing

Nama Casing

Form Factor

Brand

Price

Tambah Casing

Pilih salah satu dari menu berikut

Dengan menekan tombol Add maka akan ditampilkan sebuah *pop-up window* berisi form yang dapat diisi untuk menambahkan data baru kedalam tabel Casing.

4. Tampilan pengaturan dengan menekan tombol edit untuk metode PUT

The image shows two side-by-side screenshots of a web application window titled "Edit Tabel". Both windows have a close button (X) in the top right corner. The left window shows the form with the ID field containing the number "2". Below the ID field is a yellow "Choose" button. Further down are four empty input fields labeled "Name", "Form Factor", "Brand", and "Harga". At the bottom right of the left window is a blue "Edit!" button. The right window shows the same form but with data populated in the input fields: "Name" contains "Lian Li LANCOOL 205 MESH BLACK", "Form Factor" contains "ATX, MATX, ITX", "Brand" contains "Lian Li", and "Harga" contains "1080000". The "Choose" button is still yellow, and the "Edit!" button at the bottom right is blue.

Dengan menekan tombol Edit maka *admin* dapat melakukan pengaturan tabel berdasarkan id yang ditujukan. Ketika ID telah diisi dan menekan tombol *Choose* maka akan dilakukan query untuk mendapatkan data sesuai ID tersebut yang kemudian *valuenya* di replace menggunakan javascript sehingga akan mudah pergantian data untuk disesuaikan dengan nilai sebelumnya.

5. Tampilan untuk menghapus dengan menekan tombol Delete untuk metode DELETE

The image shows a screenshot of a web application window titled "Delete Data" with a close button (X) in the top right corner. Inside the window, there is an "ID" label above an input field. The input field has the placeholder text "Nomor ID". Below the input field, there is a small text note that says "Masukkan nomor ID casing yang akan di delete". At the bottom right of the window is a red "Delete" button.

Dengan menekan tombol Delete, maka kita dapat menspesifikan id casing yang ingin dihapus dari tabel Casing tersebut.

## Tabel Database

Pada Web App yang telah dibuat digunakan sebuah tabel yang disimpan pada database postgres dengan nama casing. Tabel ini terdiri dari sebuah *primary key* berupa *id\_case* bersifat *not null* dengan atribut lainnya berupa text dan bigint untuk menyimpan data dari casing yang akan ditampilkan pada pembeli. Berikut merupakan tampilan tabel dari database yang digunakan pada postgres:

```
sbd_webapp=# SELECT * FROM casing;
```

id_case	name	form_factor	brand	price
2	Lian Li LANCOOL 205 MESH BLACK	ATX, MATX, ITX	Lian Li	1080000
3	Corsair 4000X Tempered Glass White	ATX, MATX, ITX	Corsair	1850000
5	NZXT H510 Elite White	ATX, MATX, ITX	NZXT	2450000
6	Metallic Gear Neo Qube	ATX, MATX, ITX	Metallic Gear	2000000
7	Cube Gaming Kellva White	MATX, ITX	Cube Gaming	410000
8	Corsair Carbide 275R Airflow Tempered Glass Black	ATX, MATX, ITX	Corsair	1229000
9	MSI MPG Sekira 100R	ATX, MATX, ITX	MSI	1194000
10	PRIME X-[S] Pink - Aluminium Tempered Glass	ATX, MATX, ITX	Prime	1200000
1	Corsair 4000D Tempered Glass Black HAHA	ATX, MATX, ITX	Corsair	1379000

(9 rows)

casing	
PK	<u>id_case</u>
	name text NOT NULL
	form_factor text NOT NULL
	brand varchar(30) NOT NULL
	price bigint

Data dari tabel tersebut akan di *fetch* berdasarkan *route* yang ditujukan dari frontend dengan bantuan *middleware* berupa Express JS. Kemudian HTTPRequest yang dilakukan akan diproses sebagai query oleh NodeJS kepada database postgres. Sehingga hasil query tersebut akan diperoleh dalam bentuk response yang kemudian akan diproses oleh javascript pada frontend jika dibutuhkan.

## Bahasa Pemrograman

Dalam projek Web App tersebut digunakan beberapa bahasa pemrograman baik pada frontend maupun backend untuk membentuk sebuah aplikasi yang dapat menerapkan operasi CRUD.

Pada bagian **Frontend** digunakan tiga bahasa pemrograman untuk menunjang pembentukan *user interface* bagi pengunjung website tersebut, antara lain:

1. HTML (Hyper Text Markup Language)

Merupakan *scripting* language yang digunakan untuk membentuk struktur dari sebuah Web App. Dalam sebuah file html terdiri dari pendeskripsian konten-konten yang akan ditampilkan pada sebuah website dengan menggunakan beberapa *tag* yang tersedia. Dengan menggunakan HTML maka dalam website tersebut dapat dibentuk sebuah navigation bar, menampilkan gambar, membuat button, input, label, dan juga button. Pada setiap tag juga terdapat beberapa properti yang dapat didefinisikan seperti id yang kemudian dapat digunakan untuk membaca sebuah *event* pada Javascript seperti contoh untuk membaca ketika tombol Tampilkan ditekan yang akan memunculkan tabel. Selain itu, kolom id ini juga dapat diatur melalui CSS untuk memberikan tampilan *user interface* yang lebih menarik. Properti lainnya seperti value yang dapat mendefinisikan nilai awal pada sebuah tag *input*.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>DCS</title>
9   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
10     integrity="sha384-ggOyR0iXCbMQV3Xipma34MD+dH/1fQ784/j6cY/iJTQUohcwr7x9JvoRxT2MZw1T" crossorigin="anonymous">
11   <link rel="stylesheet" href="styles.css">
12   <link rel="icon" type="image/png" href="images/favicon.ico" />
13   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
14
15 </head>
16
17 <body>
18   <div class="container">
19     <nav class="navbar navbar-expand-lg fixed-top navbar-light">
20       <a class="navbar-brand" href="#">Darvin Casing Store</a>
21     </nav>
22   </div>
```

## 2. CSS (Cascading Style Sheets)

Merupakan *scripting* language yang digunakan untuk mengatur layout dan memberikan properti tertentu pada HTML seperti membarikan warna, peletakan, background, ukuran, dan lainnya dengan memberikan link pada stylesheetsnya. Dalam Web App ini juga digunakan sebuah Framework CSS yang dimanakan Bootstrap untuk mempermudah pendefinisian CSS pada HTML dengan menspesifikasikan class tertentu pada tag div dalam sebuah HTML. Beberapa contoh penggunaan CSS pada proyek web yang dibuat adalah peletakan dari tombol-tombol dengan menggunakan class center pada Bootstrap, membuat input dan label dengan menggunakan class form-group, dan membuat beberapa tombol yang telah diberikan properti tertentu seperti tombol Edit, Delete, dan lainnya.

```
1 .navbar{
2   background: #21c87a;
3 }
4
5 .page-banner-area {
6   padding: 150px 0px;
7 }
8
9 .page-project {
10  position: relative;
11  padding: 150px 0px;
12  background: url("images/banner/p-4.jpg") fixed 10% 10% no-repeat;
13  background-size: cover;
14 }
15
16 .overlay {
17  position: absolute;
18  left: 0;
19  top: 0;
20  width: 100%;
21  height: 100%;
22  background: rgba(0, 0, 0, 0.92);
23 }
```

## 3. Javascript

Merupakan bahasa pemrograman untuk membentuk website yang lebih interaktif dan dapat digunakan untuk menjalankan fungsi dan instruksi tertentu terhadap halaman website berdasarkan htmlnya. Dengan menggunakan JS maka dapat dilakukan juga

pemanggilan fungsi tertentu berdasarkan *event* yang terjadi pada objek tertentu dalam html. Pada web ini juga digunakan library JS berupa JQuery untuk mempermudah penulisan sintaks tertentu sehingga akan mempermudah pemrogramannya. Beberapa penggunaan Javascript pada proyek DCS adalah untuk membaca event *submit* pada form penambahand ata yang kemudian akan melakukan metode POST pada backend, mengganti properti dalam sebuah elemen untuk digantikan dengan data dinamis pada Edit data dengan operasi GET, dan menampilkan tabel ketika event *click* dengan operasi.

```
28 $('#chooseID').on('click', function (event) {
29     // console.log("Hello World");
30     var inputVal = document.getElementById("editID").value;
31     console.log(inputVal);
32     $.getJSON('http://localhost:3080/editID/${inputVal}', function (data) {
33         document.getElementById("editName").setAttribute('value', data[0].name);
34         document.getElementById("editFF").setAttribute('value', data[0].form_factor);
35         document.getElementById("editBrand").setAttribute('value', data[0].brand);
36         document.getElementById("editPrice").setAttribute('value', data[0].price);
37     });
38 });
39
40 //Mengirim data baru yang akan dimasukkan dengan POST
41 const thisForm = document.getElementById('myForm');
42 thisForm.addEventListener('submit', async function (e) {
43     e.preventDefault();
44     const formData = new FormData(thisForm).entries()
45     const response = await fetch('http://localhost:3080/tambah', {
46         method: 'POST',
47         headers: { 'Content-Type': 'application/json' },
48         body: JSON.stringify(Object.fromEntries(formData))
49     });
50
51     const result = await response.json();
52     console.log(result)
53 });
```

Sedangkan pada bagian **Backend** juga digunakan bahasa pemrograman untuk memproses database, antara lain:

1. SQL (Structured Query Language)

Merupakan sebuah bahasa pemrograman yang ditujukan untuk memproses sebuah *relational database* dengan melakukan *query*. Dengan menggunakan SQL maka dapat dilakukan beberpa operasi seperti INSERT untuk memasukkan data, CREATE untuk membuat tabel baru, UPDATE untuk mengganti nilai data tertentu, Delete untuk menghapus suatu tabel, dan instruksi lainnya. Pada Web App ini dengan adanya SQL maka akan mempermudah penampilan data dari database dengan mengirim sebuah HttpRequest untuk melakukan query tertentu pada database sehingga diperoleh data atau respond yang sesuai.

```

4 CREATE TABLE casing
5 (id_case serial PRIMARY KEY NOT NULL,
6 name text NOT NULL,
7 form_factor text NOT NULL,
8 brand varchar(30) NOT NULL,
9 price bigint
10 );
11
12 INSERT INTO casing (name, form_factor, brand, price) VALUES (
13 'Corsair 4000D Tempered Glass Black', 'ATX, MATX, ITX', 'Corsair', 1379000),(
14 'Lian Li LANCOOL 205 MESH BLACK', 'ATX, MATX, ITX', 'Lian Li', 1080000),(
15 'Corsair 4000X Tempered Glass White', 'ATX, MATX, ITX', 'Corsair', 1850000),(
16 'Corsair 465X RGB Tempered Glass Black', 'ATX, MATX, ITX', 'Corsair', 2279000),(
17 'NZXT H510 Elite White', 'ATX, MATX, ITX', 'NZXT', 2450000),(
18 'Metallic Gear Neo Qube', 'ATX, MATX, ITX', 'Metallic Gear', 2000000),(
19 'Cube Gaming Kellva White', 'MATX, ITX', 'Cube Gaming', 410000),(
20 'Corsair Carbide 275R Airflow Tempered Glass Black', 'ATX, MATX, ITX', 'Corsair', 1229000),(
21 'MSI MPG Sekira 100R', 'ATX, MATX, ITX', 'MSI', 1194000),(
22 'PRIME X-[S] Pink - Aluminium Tempered Glass', 'ATX, MATX, ITX', 'Prime', 1200000);

```

## 2. Javascript (Node.JS)

Selain digunakan pada Frontend, JS juga dapat digunakan sebagai Backend dengan menggunakan API node.js. Penggunaan javascript di backend tersebut ditujukan untuk mendefinisikan beberapa route. Route tersebut kemudian akan dipanggil dari *frontend* untuk menjalankan query tertentu pada database. Dalam hal ini digunakan sebuah middleware berupa ExpressJS untuk menghubungkan antara frontend dengan backend. Dalam Web App tersebut dengan adanya route tersebut maka dapat dibuka sebuah website dinamis yang mengambil data dari database dan dapat memproses penghapusan dan pengaturan data dari *frontend*nya.

```

23 < client.connect((err) => {
24 <   if (err) {
25 <     console.error(err);
26 <     return;
27 <   }
28 <   console.log('Database Connected');
29 < });
30
31 //Create//
32 < app.post('/tambah', function (req, res) {
33 <   const query = // Gunakan query yang sesuai
34 <   `INSERT INTO casing (name, form_factor, brand, price) VALUES (
35 <     '${req.body.name}', '${req.body.form_factor}', '${req.body.brand}',
36 <     '${req.body.price}')`;
37 <   client.query(query, (err, results) => {
38 <     if (err) {
39 <       console.error(err);
40 <       return;
41 <     }
42 <     res.send('Data Casing berhasil diubah');
43 <   });
44 < });

```