

PRAKTIKUM PEMROGRAMAN WEB



MANUAL BOOK

**“Aplikasi Data Diri Penjualan Untuk Mencatat Jadwal Ujian Praktikum
Menggunakan Visual Studio Code, Html, CSS dan JavaScript”**

Nama : Ahmad Thohir Hilmi
Npm : 50421081
Kelas : 3IA09
Fakultas : Teknologi Industri
Jurusan : Informatika
PJ : Dariwan

**Ditulis Guna Melengkapi Sebagian Syarat Praktikum Pemrograman Web
Jenjang S1 Universitas Gunadarma 2023**

BAB I PENDAHULUAN

1.1 LATAR BELAKANG

Pertumbuhan teknologi informasi, terutama di bidang pengembangan web, telah menciptakan kebutuhan yang terus meningkat untuk aplikasi web yang interaktif dan responsif. Dalam konteks ini, operasi CRUD (Create, Read, Update, Delete) menjadi bagian krusial dari pengembangan web.

CRUD merangkum empat operasi dasar yang diperlukan untuk mengelola data pada suatu aplikasi.

Penggunaan bahasa pemrograman JavaScript, HTML, dan CSS dalam pengembangan web semakin mendominasi karena kemampuannya untuk membuat antarmuka yang dinamis dan menarik. JavaScript sebagai bahasa pemrograman di sisi klien memberikan kekuatan untuk berinteraksi dengan pengguna dan mengelola data secara real-time. HTML digunakan untuk struktur dasar dokumen web, sementara CSS memberikan gaya dan tata letak yang estetik.

Makalah ini bertujuan untuk mengeksplorasi perancangan aplikasi web berbasis CRUD menggunakan JavaScript, HTML, dan CSS. Implementasi CRUD pada aplikasi web menjadi penting karena memberikan kemampuan untuk membuat, membaca, memperbarui, dan menghapus data secara efisien, yang merupakan aspek krusial dalam pengelolaan informasi.

1.2 TUJUAN

Melalui pemahaman yang mendalam tentang konsep CRUD dan penerapannya menggunakan bahasa JavaScript, HTML, dan CSS, makalah ini diharapkan dapat memberikan wawasan yang berharga bagi pengembang web dan mahasiswa yang tertarik pada pengembangan aplikasi web modern. Dengan merangkul teknologi ini, kita dapat menciptakan aplikasi web yang dinamis, responsif, dan efisien dalam pengelolaan data.

BAB II PEMBAHASAN

2.1 VISUAL STUDIO CODE

Visual studio Code merupakan aplikasi cross platform yang dapat digunakan berbagai sistem operasi seperti windows, Linux, dan Mac OS. VS Code termasuk software yang ringan namun kuat editor sumbernya dengan deskop. Menggunakan berbagai macam bahasa pemrograman seperti Java, JavaScript, Go, C++, dan masih banyak yang lainnya. Komponen dari Visual Studio juga sama seperti yang digunakan di Azure DevOps. Visual Studio memiliki lintas platform kode editor yang ringan, dapat digunakan oleh siapa saja untuk membuat atau membangun aplikasi web.:

2.1.1 KELEBIHAN VISUAL STUDIO CODE

- Mudah untuk mengelola extension
- Memiliki extension yang banyak
- Kontribusi tampilan
- Dukungan bahasa
- Text editor Gratis
- Dapat membuat Snippet sendiri
- Mudah dalam memahami dan mempelajari coding
- Dapat membuat website secara praktis
- Dapat mendesain website dengan mudah

2.1.2 KEKURANGAN VISUAL STUDIO CODE

Aplikasi VSCode juga memiliki kekurangan dalam pengembangannya, diantaranya:

1. Belum bisa mendeteksi secara otomatis bahasa apa yang sedang digunakan
2. Aplikasinya lumayan berat saat dijalankan

2.2 CRUD

CRUD adalah singkatan dari Create, Read, Update, dan Delete. Keempat hal tersebut merupakan perintah dengan peran yang esensial dalam sebuah aplikasi database yang sifatnya relational. Beberapa contoh DBMS (Database Management System) yang menggunakan CRUD sebagai fungsi utamanya yaitu Oracle Database, Microsoft SQL Server, MySQL, PostgreSQL, dan sebagainya.

2.2.1 PENERAPAN KONSEP CRUD

Secara singkat, hampir semua aplikasi atau website yang menggunakan *relational database* sangat bergantung pada fungsionalitas CRUD. Tak hanya itu, segala bentuk aplikasi yang mengizinkan pengguna untuk membuat akun juga bergantung pada CRUD atau bentuk lainnya seperti:

- BREAD (Browse, Read, Edit, Add, Delete)
- CRAP (Create, Replicate, Append, dan Process)
- CRUDL (Create, Read, Update, Delete, Lis(t))
- DAVE (Delete, Add View, Edit)

2.2.2 KELEBIHAN CRUD

- Digunakan oleh berbagai bahasa pemrograman
- Operasi Pemrograman yang Cukup Kompleks
- Mendukung Tampilan UI Berjalan dengan Baik

2.3 JavaScript

JavaScript adalah bahasa pemrograman yang digunakan developer untuk membuat halaman web yang interaktif. Dari menyegarkan umpan media sosial hingga menampilkan animasi dan peta interaktif, fungsi JavaScript dapat meningkatkan pengalaman pengguna situs web. Sebagai bahasa skrip sisi klien, JavaScript adalah salah satu teknologi inti dari World Wide Web. Misalnya, saat menjelajah internet, kapan pun Anda melihat carousel gambar, menu tarik-turun klik untuk menampilkan, atau warna elemen yang berubah secara dinamis di halaman web, Anda melihat efek JavaScript.

Secara historis, halaman web statis, mirip dengan halaman dalam buku. Halaman statis terutama menampilkan informasi dalam tata letak tetap dan tidak melakukan semua yang kita harapkan saat ini dari situs web modern. JavaScript muncul sebagai teknologi sisi peramban untuk menjadikan aplikasi web lebih dinamis. Dengan JavaScript, browser dapat merespons interaksi pengguna dan mengubah tata letak konten di halaman web.

Ketika bahasanya sudah mapan, developer JavaScript membuat pustaka, kerangka kerja, serta praktik pemrograman, dan mulai menggunakannya di luar browser web.

2.3.1 HTML

Hypertext Markup Language atau HTML adalah bahasa markup standar yang digunakan untuk membuat halaman website dan aplikasi web.

Sejarah HTML dimulai oleh Tim Berners-Lee, seorang ahli fisika di lembaga penelitian CERN yang berlokasi di Swiss. Versi pertamanya dirilis pada tahun 1991, dengan 18 tag. Sejak saat itu, setiap kali ada versi barunya, pasti akan selalu ada tag dan attribute (tag modifier) yang juga baru.

HTML disusun berdasar kode dan simbol tertentu, yang dimasukkan dalam sebuah file atau dokumen. Sehingga bisa ditampilkan pada layar komputer. Dan bisa dipahami oleh para pengguna internet.

Memahami setiap kata yang terkandung, hypertext sendiri dimaksudkan sebagai metode yang digunakan untuk berpindah laman web ke laman lain. Usai mengklik tulisan atau simbol yang muncul di halaman website.

2.3.2 CSS

CSS adalah singkatan dari *cascading style sheets*, yaitu bahasa yang digunakan untuk menentukan tampilan dan format halaman website. Dengan CSS, Anda bisa mengatur jenis font, warna tulisan, dan latar belakang halaman.

CSS digunakan bersama dengan bahasa markup, seperti HTML dan [XML](#) untuk membangun sebuah website yang menarik dan memiliki fungsi yang berjalan baik.

CSS juga berguna untuk mengatasi keterbatasan HTML dalam mengatur format halaman website. Kenapa demikian?

Apabila hanya menggunakan HTML ketika membangun website dengan beberapa halaman, Anda harus menulis tag untuk sebuah elemen HTML di semua halaman tersebut.

CSS sangat mudah dipelajari, tapi juga powerful karena dapat mengontrol penyajian tampilan dari dokumen HTML. Mulai dari yang simpel sampai kompleks. Tidak heran jika saat ini CSS hampir dipakai di berbagai website untuk dikombinasikan dengan HTML maupun PHP

BAB III

ANALISA DAN PERANCANGAN

1. Pertama, melakukan install visual studio code dengan cara mendownload di website visual studio code <https://code.visualstudio.com/> sesuai versi yang dibutuhkan, setelah itu tunggu hingga selesai instalasi.
2. Kedua, membuka folder dari project yang ingin kalian buat
3. Lalu membuat file index.html, script.js dan style.css
4. Kemudian kita membuat kodingan untuk index.html sebagai kerangka website kita seperti berikut :

Index.html

```
<!DOCTYPE html>
<html>

<head>
  <title>
    JavaScript CRUD Example Tutorial
  </title>
  <link rel="stylesheet" href="style.css">
</head>

<body>

  <h1><center>Data Diri Penjualan</center></h1>
  <hr>

    <div class="employee-form">
      <form onsubmit="event.preventDefault();onFormSubmit();"
autocomplete="off">
        <div>
          <label>Nama*</label><label class="validation-error hide"
id="fullNameValidationError">This field is required.</label>
          <input type="text" name="Nama" id="fullName">
        </div>
        <div>
          <label>Email</label>
          <input type="text" name="Email" id="email">
        </div>
        <div>
          <label>Penjualan</label>

```

```

        <input type="text" name="Penjual" id="penjualan">
    </div>
    <div>
        <label>Kota</label>
        <input type="text" name="Kota" id="kota">
    </div>
    <div class="form-action-buttons">
        <input type="submit" value="Submit">
    </div>
</form>
</div>
<br/>
<div class = "employees-table">
    <table class="list" id="employeeList">
        <thead>
            <tr>
                <th>Nama</th>
                <th>Email </th>
                <th>Penjualan</th>
                <th>Kota</th>
                <th>Edit</th>
            </tr>
        </thead>
        <tbody>

        </tbody>
    </table>
</div>
<script src="script.js"></script>
</body>
</html>

```

Penjelasan :

Kode HTML ini membangun halaman web sederhana dengan struktur dasar.

Bagian <head> berisi informasi meta dan link ke file CSS untuk styling.

Bagian <body> menampung konten utama halaman, termasuk:

Judul halaman: "Data Diri Penjualan"

Elemen <hr> untuk garis pembatas.

Div <employee-form> yang berisi formulir penambahan data.

Div <employees-table> yang akan menampilkan daftar data dalam tabel.

Skrip JavaScript script.js untuk menangani logika aplikasi.

Formulir Penambahan Data:

Formulir memiliki atribut onsubmit untuk menjalankan fungsi onFormSubmit() ketika formulir di-submit.

Formulir berisi input untuk:

Nama (wajib diisi)

Email

Penjualan

Kota

Elemen label dan input dibungkus dalam <div> untuk pengorganisasian.

Label untuk Nama memiliki elemen <label class="validation-error hide" id="fullNameValidationError"> yang akan menampilkan pesan kesalahan jika nama tidak diisi.

Tombol submit untuk mengirimkan data formulir.

Tabel Daftar Data:

Tabel memiliki ID "employeeList" untuk memudahkan manipulasi JavaScript.

Header tabel berisi kolom untuk:

Nama

Email

Penjualan

Kota

Edit

Badan tabel (tbody) masih kosong, data akan ditambahkan secara dinamis menggunakan JavaScript.

Scripts.js

```
var selectedRow = null

function onFormSubmit() {
    if (validate()) {
        var formData = readFormData();
        if (selectedRow == null)
            insertNewRecord(formData);
        else
            updateRecord(formData);
        resetForm();
    }
}

function readFormData() {
    var formData = {};
    formData["nama"] = document.getElementById("fullName").value;
    formData["email"] = document.getElementById("email").value;
    formData["penjualan"] = document.getElementById("penjualan").value;
    formData["kota"] = document.getElementById("kota").value;
    return formData;
}

function insertNewRecord(data) {
    var table =
document.getElementById("employeeList").getElementsByName('tbody')[0];
    var newRow = table.insertRow(table.length);
    cell1 = newRow.insertCell(0);
    cell1.innerHTML = data.nama;
    cell2 = newRow.insertCell(1);
    cell2.innerHTML = data.email;
    cell3 = newRow.insertCell(2);
    cell3.innerHTML = data.penjualan;
    cell4 = newRow.insertCell(3);
    cell4.innerHTML = data.kota;
    cell4 = newRow.insertCell(4);
    cell4.innerHTML = `
```

```

function onEdit(td) {
    selectedRow = td.parentElement.parentElement;
    document.getElementById("nama").value = selectedRow.cells[0].innerHTML;
    document.getElementById("email").value = selectedRow.cells[1].innerHTML;
    document.getElementById("penjualan").value = selectedRow.cells[2].innerHTML;
    document.getElementById("kota").value = selectedRow.cells[3].innerHTML;
}
function updateRecord(formData) {
    selectedRow.cells[0].innerHTML = formData.fullName;
    selectedRow.cells[1].innerHTML = formData.email;
    selectedRow.cells[2].innerHTML = formData.penjualan;
    selectedRow.cells[3].innerHTML = formData.kota;
}

function onDelete(td) {
    if (confirm('Are you sure to delete this record ?')) {
        row = td.parentElement.parentElement;
        document.getElementById("employeeList").deleteRow(row.rowIndex);
        resetForm();
    }
}

function validate() {
    isValid = true;
    if (document.getElementById("fullName").value == "") {
        isValid = false;
        document.getElementById("fullNameValidationError").classList.remove("hide");
    } else {
        isValid = true;
        if
(!document.getElementById("fullNameValidationError").classList.contains("hide"))
            document.getElementById("fullNameValidationError").classList.add("hide");
    }
    return isValid;
}

```

Penjelasan :

selectedRow: Variabel global untuk menyimpan baris yang sedang dipilih pada tabel.

onFormSubmit: Fungsi untuk menangani submit form.

readFormData: Fungsi untuk membaca data dari form.

insertNewRecord: Fungsi untuk menambahkan baris baru ke tabel.

resetForm: Fungsi untuk mereset formulir.

onEdit: Fungsi untuk mengedit baris yang dipilih.

updateRecord: Fungsi untuk memperbarui data pada baris yang dipilih.

onDelete: Fungsi untuk menghapus baris yang dipilih.

validate: Fungsi untuk memvalidasi formulir.

2. Penjelasan Fungsi:

onFormSubmit:

Memanggil validate untuk memeriksa validitas form.

Jika valid, panggil readFormData untuk mendapatkan data dari form.

Jika selectedRow null, panggil insertNewRecord untuk menambahkan data baru.

Jika tidak, panggil updateRecord untuk memperbarui data.

Panggil resetForm untuk mereset formulir.

readFormData:

Membaca data dari form dan menyimpannya ke objek formData.

Mengembalikan objek formData.

insertNewRecord:

Mendapatkan elemen tabel dan tbody.

Memasukkan baris baru ke akhir tabel.

Mengisi sel-sel baris baru dengan data dari formData.

Menambahkan link "Edit" dan "Delete" ke sel terakhir.

resetForm:

Mengosongkan semua input pada formulir.

Mengatur selectedRow menjadi null.

onEdit:

Mendapatkan baris yang sedang dipilih dengan mengambil parentElement dari td yang diklik.

Mengisi formulir dengan data yang terdapat pada sel-sel baris yang dipilih.

Mengatur selectedRow dengan baris yang dipilih.

updateRecord:

Mengisi sel-sel baris yang dipilih dengan data dari formData.

onDelete:

Meminta konfirmasi sebelum menghapus baris.

Jika dikonfirmasi, hapus baris yang dipilih dari tabel.

Panggil resetForm untuk mereset formulir.

validate:

Memeriksa apakah input "Nama" kosong.

Jika kosong, set isValid menjadi false dan tampilkan pesan kesalahan.

Jika tidak, set isValid menjadi true dan sembunyikan pesan kesalahan.

Mengembalikan isValid.

Style.css

```
.employee-form {
    border-style: solid;
    /* margin-bottom: 10px; */
    /* margin-left: 10px; */
    padding: 10px;
    /* width: 50%; */
    margin: auto;
    width: 50%;
    /* border: 3px solid green; */
    /* padding: 10px; */
}

.employees-table {
    border-style: solid;
    /* margin-bottom: 10px; */
    /* margin-left: 10px; */
    padding: 20px;
    /* width: 50%; */
    margin: auto;
    width: 70%;
    /* border: 3px solid green; */
    /* padding: 10px; */
}

body > table{
    width: 80%;
}

table{
    border-collapse: collapse;
}

table.list{
    width:100%;
}

td, th {
    border: 1px solid #dddddd;
    text-align: left;
    padding: 8px;
}

tr:nth-child(even),table.list thead>tr {
    background-color: #dddddd;
}

input[type=text], input[type=number] {
    width: 100%;
```

```

padding: 8px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
}

input[type=submit] {
width: 30%;
background-color: black;
color: white;
padding: 10px 18px;
/* margin: 0px 0; */
border: none;
border-radius: 5px;
cursor: pointer;
}

form div.form-action-buttons{
text-align: right;
}

a{
cursor: pointer;
text-decoration: underline;
color: #0000ee;
margin-right: 4px;
}

label.validation-error{
color: red;
margin-left: 5px;
}

.hide{
display:none;
}

```

Penjelasan :

.employee-form:

Menerapkan garis tepi (border-style: solid) pada elemen dengan kelas .employee-form.
 Memberikan padding 10px untuk memberikan ruang di sekitar isi elemen.
 Menengahkan elemen dengan margin: auto, sehingga form berada di tengah halaman.
 Menetapkan lebar elemen menjadi 50% dari lebar halaman.

.employees-table:

Menerapkan garis tepi (border-style: solid) pada elemen dengan kelas .employees-table.

Memberikan padding 20px untuk memberikan ruang di sekitar isi elemen.

Menengahkan elemen dengan margin: auto, sehingga tabel berada di tengah halaman.

Menetapkan lebar elemen menjadi 70% dari lebar halaman.

body > table:

Menetapkan lebar tabel tingkat atas yang langsung berada di bawah elemen body menjadi 80% dari lebar halaman.

table:

Menetapkan properti untuk menggabungkan tepi sel (border-collapse: collapse) pada semua tabel.

table.list:

Menetapkan lebar tabel dengan kelas .list menjadi 100% dari lebar elemen induknya.

td, th:

Menetapkan properti batasan sel dan heading tabel, seperti tepi dan padding.

Menerapkan warna latar belakang abu-abu pada baris genap dan pada heading tabel untuk membedakan baris.

tr:nth-child(even),table.list thead>tr:

Memberikan warna latar belakang abu-abu pada baris genap pada tabel dan heading tabel untuk membedakan baris.

input[type=text], input[type=number]:

Menetapkan lebar 100% untuk input teks dan angka.

Memberikan padding dan margin untuk penataan dan tampilan yang baik.

Menetapkan tepi dan border-radius untuk tampilan input yang bersih dan ramping.

input[type=submit]:

Menetapkan lebar 30% dari lebar elemen induknya untuk tombol submit.

Memberikan warna latar hitam dan warna teks putih pada tombol.

Memberikan padding untuk tampilan yang baik.

Menetapkan border-radius agar tombol memiliki sudut yang melengkung.

Menetapkan pointer cursor saat digunakan.

form div.form-action-buttons:

Menetapkan penataan teks ke kanan untuk elemen div dengan kelas .form-action-buttons.

a:

Menetapkan pointer cursor pada tautan (link).

Memberikan efek underline dan warna biru pada tautan.

Memberikan margin kanan 4px untuk memberikan ruang antar tautan.

label.validation-error:

Menetapkan warna teks merah pada label dengan kelas .validation-error.

Menetapkan margin kiri 5px untuk memberikan ruang antara teks dan elemen sekitarnya

Output :

Data Diri Penjualan

Nama*

Jamal

Email

Jamal@gmail.com

Penjualan

Celana

Kota

Depok

Submit

Nama	Email	Penjualan	Kota	Edit
Hilmi	ahmmadhilmi78@gmail.com	baju	Bogor	Edit Delete
Jamal	Jamal@gmail.com	Celana	Depok	Edit Delete

Data Diri Penjualan

Nama*

Jamal

Email

Jamal@gmail.com

Penjualan

Celana

Kota

Depok

Submit

Nama	Email	Penjualan	Kota	Edit
Hilmi	ahmmadhilmi78@gmail.com	baju	Bogor	Edit Delete
Jamal	Jamal@gmail.com	Celana	Depok	Edit Delete

BAB IV

PENUTUP

4.1 Kesimpulan

Dalam perancangan CRUD menggunakan bahasa JavaScript, HTML, dan CSS berbasis website, dapat disimpulkan bahwa pendekatan ini memberikan kerangka kerja yang kuat untuk mengembangkan aplikasi web yang interaktif dan dinamis. Dengan memanfaatkan kekuatan JavaScript untuk logika bisnis, HTML untuk struktur dokumen, dan CSS untuk tata letak dan gaya, pengembang dapat membuat antarmuka pengguna yang menarik dan efisien. Penerapan operasi CRUD (Create, Read, Update, Delete) memungkinkan manajemen data yang efektif, baik untuk penyimpanan lokal maupun pengelolaan data pada server. Formulir input memfasilitasi pengguna untuk menambahkan dan mengedit data dengan mudah, sementara tampilan tabular memberikan visibilitas yang baik terhadap dataset.

Penggunaan Bootstrap sebagai framework front-end menyediakan gaya bawaan, komponen UI yang siap pakai, dan responsivitas yang diperlukan untuk mendukung pengembangan aplikasi web modern. Dengan responsivitasnya, aplikasi ini dapat diakses dengan nyaman melalui berbagai perangkat, dari desktop hingga perangkat mobile.

Pentingnya penyimpanan data menggunakan local storage atau integrasi dengan server memberikan fleksibilitas dalam mengelola data, tergantung pada kebutuhan proyek. Keseluruhan, perancangan CRUD dengan JavaScript, HTML, dan CSS membuka pintu bagi pengembangan aplikasi web yang tangguh, dengan kemampuan untuk menyediakan pengalaman pengguna yang dinamis dan responsif. Seiring dengan terus berkembangnya teknologi web, pendekatan ini tetap relevan dan memberikan dasar yang kokoh bagi pengembangan aplikasi modern yang memenuhi tuntutan pasar dan kebutuhan pengguna.

4.2 Saran

Pertimbangkan untuk mendalami aspek keamanan, terintegrasi dengan backend untuk skalabilitas, responsivitas desain untuk pengalaman pengguna yang lebih baik, optimalkan kinerja dan kelola memori, dokumentasikan kode dengan baik, uji perangkat lunak secara efektif, pertimbangkan penggunaan database, bandingkan dengan teknologi lain, dan diskusikan pemeliharaan serta pengembangan aplikasi ke depan.