

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS WEB
“CRUD (*Create, Read, Update, Delete*)”



Disusun Oleh:

Nama : Theresia Seftiani Girsang

Nim : 1808561029

Kelas : B

PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA
BADUNG
2020

KATA PENGANTAR

Puji dan syukur ke hadirat Tuhan Yang Maha Kuasa atas segala rahmat yang diberikan-Nya sehingga tugas Laporan Praktikum Pemrograman Berbasis Web yang berjudul “CRUD (*Create, Read, Update, Delete*)” ini dapat saya selesaikan. Laporan ini saya buat sebagai kewajiban untuk memenuhi tugas. Dalam kesempatan ini, penulis menghaturkan terimakasih yang dalam kepada semua pihak yang telah membantu menyumbangkan ide dan pikiran mereka demi terwujudnya laporan ini. Akhirnya saran dan kritik pembaca yang dimaksud untuk mewujudkan kesempurnaan laporan ini penulis sangat hargai.

Badung, Desember 2020

Theresia Seftiani Girsang

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan.....	1
1.3 Manfaat.....	1
BAB II KAJIAN PUSTAKA	2
2.1 CRUD	2
2.2 User Interface	2
BAB III PEMBAHASAN	4
3.1 Tugas Praktikum Pertemuan 8.....	4
3.2 Pembahasan Tugas.....	4
BAB IV PENUTUP	12
4.1 Simpulan.....	12
4.2 Saran.....	12
DAFTAR PUSTAKA.....	13

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem informasi yang baik akan menyimpan dengan baik setiap proses transaksi yang terjadi ke dalam database. Hal ini menjadi hal yang krusial, karena ini menjadi titik vital dari perusahaan, karena setiap transaksi perusahaan tercatat dengan baik. Untuk menunjang itu diperlukannya penggunaan database yang baik.

Dalam proses database saat ini sering menggunakan database relational atau RDBMS. Proses database ini menggunakan query untuk memanggil data dari database. Sebuah sistem informasi akan sering melakukan proses query ke dalam database, hal ini untuk mendapatkan data yang sesuai dan dalam keadaan nyata. Untuk itu diperlukan sebuah proses query database yang baik.

Proses CRUD ini akan menjadi sangat penting karena, akan memberatkan sistem jika proses berlangsung secara serampangan. Salah satu prosesnya adalah teknik normalisasi dari database, teknik ini sudah sering digunakan dan menjadi bagian penting. Salah satu teknik penting di dalam proses CRUD ke database, kita bisa menggunakan teknik ORM atau Object Relational Mapping, terlebih teknik ini akan sangat bekerja optimal dibawah pengaruh Object Oriented.

1.2 Tujuan

1. Memfasilitasi mahasiswa dalam memahami konsep dasar CRUD.
2. Memfasilitasi mahasiswa dalam memahami cara implementasi CRUD.

1.3 Manfaat

1. Mengetahui dan memahami konsep dasar CRUD.
2. Mengetahui dan memahami cara penggunaan CRUD pada sistem.

BAB II

KAJIAN PUSTAKA

2.1 CRUD

CRUD merupakan empat fungsi dasar dari penyimpanan persisten, diantaranya:

- C (*Create*), berarti membuat sebuah data baru, contoh kita sedang melakukan registrasi di sebuah web itu sudah merupakan Create dari CRUD karena kita membuat dan menyimpan data registrasi ke database.
- R (*Read*), membaca atau menampilkan suatu data yang tadinya berada di database MySQL misalnya, kemudian ditampilkan di WEB menggunakan bahasa pemrograman Php.
- U (*Update*), adalah mengedit sebuah data dari database yang kemudian di edit menggunakan bahasa pemrograman Php berupa Website.
- D (*Delete*), fungsinya hampir sama dengan Update akan tetapi proses ini adalah untuk melakukan penghapusan data di database melalui bahasa Php.

2.2 User Interface

CRUD juga relevan di tingkat user interface dari sebagian besar aplikasi. Misalnya, dalam buku alamat software, unit penyimpanan dasar adalah seorang individu kontak entri. Sebagai minimal, perangkat lunak harus memungkinkan pengguna untuk :

1. Membuat atau menambah entri baru.
2. Baca, mengambil, pencarian atau melihat entri yang ada.
3. Memperbaharui atau mengedit entri yang ada.
4. Hapus/menonaktifkan/menghapus entri yang ada.

Tanpa empat operasi setidaknya ini, perangkat lunak tidak dapat dianggap lengkap. Karena operasi ini sangat mendasar, mereka sering didokumentasikan dan dijelaskan di bawah satu judul yang luas, seperti "manajemen kontak", "manajemen konten" atau "pemeliharaan kontak"

(atau "manajemen dokumen" secara umum, tergantung pada unit penyimpanan dasar untuk aplikasi tertentu).

BAB III

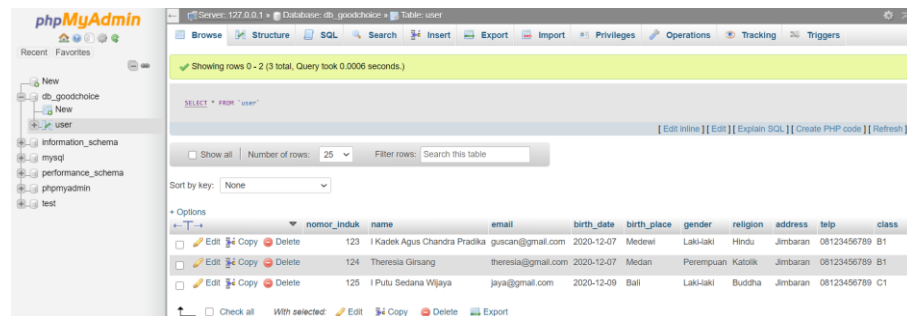
PEMBAHASAN

3.1 Tugas Praktikum Pertemuan 8

Buatlah sebuah program CRUD dari final project yang kalian kerjakan.

3.2 Pembahasan Tugas

- Siapkan database yang akan diolah dan text editor untuk membuat script code CRUD nya.



- Tampilan dan penjelasan code config.php

```
C: > xampp > htdocs > Praktikum 8 > Config.php
1  <?php
2
3  $server = "localhost";
4  $user = "root";
5  $password = "";
6  $nama_database = "db_goodchoice";
7
8  $db = mysqli_connect($server, $user, $password, $nama_database);
9
10 if (!$db) {
11     die("Gagal terhubung dengan database: " . mysqli_connect_error());
12 }
```

Penjelasan:

Code config ini berfungsi untuk menghubungkan database dengan website dengan menggunakan function `mysqli_connect()` yang parameternya adalah data dari server pada xampp dan database. Jika tidak berhasil terhubung maka akan muncul error. File config.php ini akan dipanggil di setiap code pemrosesan database CRUD.

- c. Tampilan dan penjelasan code create pada data_insert.php dan proses_insert.php

```
<!-- Start Navbar -->
<nav class="navbar navbar-expand-lg navbar-light bg-transparent border-bottom" data-aos="fade-right" data-aos-duration="2000">
  <div class="container">
    <a class="navbar-brand" href="#"><i class="fas fa-book-reader"></i>&nbsp;Good Choice</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="index.php?Home"><span class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="data_insert.php?Tambah Data"></a>
        </li>
        <li class="nav-item">
          <a class="btn btn-block btn-outline-danger" href="index.php?KELUAR"></a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Penjelasan:

Di dalam tag body berisi navbar untuk tampilan bar di atas yang menampilkan logo, tombol Home yang mengarah ke index.php, tombol Tambah Data yang mengarah ke file ini, dan Tombol keluar yang saat ini mengarah ke index.php.

```
<div class="jumbotron jumbotron-fluid search-wrapper bg-dark">
  <h1 class="display-4 text-center text-white" data-aos="fade-down" data-aos-duration="2000">Tambah Data<br>Sistem Informasi Good Choice</h1>
</div>
<!-- End Navbar -->
<!-- Start Form -->
<div class="container" data-aos="fade-up" data-aos-duration="2000">
  <form action="proses_insert.php" method="POST">
    <div class="form-group">
      <label for="nomor_induk">Nomor Induk</label>
      <input type="text" name="nomor_induk" class="form-control" id="nomor_induk">
    </div>
    <div class="form-group">
      <label for="nama">Nama</label>
      <input type="text" name="nama" class="form-control" id="nama">
    </div>
    <div class="form-group">
      <label for="email">Email</label>
      <input type="email" class="form-control" name="email" id="email">
    </div>
    <div class="form-group">
      <label for="birth_date">Tanggal Lahir</label>
      <input type="date" class="form-control" name="birth_date">
    </div>
    <div class="form-group">
      <label for="birth_place">Tempat Lahir</label>
      <input type="text" class="form-control" name="birth_place" id="birth_place">
    </div>
    <div class="form-group">
      <label for="gender">Jenis kelamin</label>
      <select class="form-control" name="gender">
        <option value="laki-laki">Laki-laki</option>
        <option value="perempuan">Perempuan</option>
      </select>
    </div>
  </form>
</div>
```

Penjelasan:

Kemudian terdapat jumbotron yang menampilkan informasi judul, dan formulir yang berisi nomor induk, nama, email, tanggal lahir, tempat lahir, jenis kelamin, agama, alamat, no. telp, kelas, dan tombol submit untuk user menginputkan data yang akan di create pada database dengan value unik dari masing – masing data. Terakhir berisi code footer yang menampilkan copyright pemilik website. (code tidak ditampilkan keseluruhan karena cukup panjang).


```

1 // cek apakah tombol submit sudah diklik atau tidak
2 <input type="button" value="Tambah Data" />
3
4 // cek apakah tombol submit sudah diklik atau tidak
5 if (isset($_POST['tambah'])) {
6     // ambil data dari form
7     $nomor_induk = $_POST['nomor_induk'];
8     $nama = $_POST['nama'];
9     $email = $_POST['email'];
10    $birth_date = $_POST['birth_date'];
11    $birth_place = $_POST['birth_place'];
12    $gender = $_POST['gender'];
13    $religion = $_POST['religion'];
14    $address = $_POST['address'];
15    $telepon = $_POST['telepon'];
16    $kelas = $_POST['kelas'];
17
18    // buat query insert
19    $sql = "INSERT INTO user (nomor_induk, nama, email, birth_date, birth_place, gender, religion, address, telepon, kelas) VALUES ('$nomor_induk','$nama','$email','$birth_date','$birth_place','$gender','$religion','$address','$telepon','$kelas')";
20    $query = mysqli_query($db, $sql);
21
22    // cek apakah query berhasil
23    if ($query) {
24        // jika berhasil alihkan ke halaman index.php dengan status sukses
25        header('location: index.php?status=sukses');
26    } else {
27        // jika gagal alihkan ke halaman index.php dengan status gagal
28        header('location: index.php?status=gagal');
29    }
30 } else {
31     die("Akses dilarang...");
32 }

```

Penjelasan:

Saat user mengklik tombol submit, nilai yang dikirim adalah seluruh data yang ada pada formulir ke variabel `$_POST`. Oleh itu pada proses insert dilakukan terlebih dahulu apakah user mengklik tombol submit dengan pengkondisian `isset($_POST['tambah'])`. Jika user telah mengklik tombol submit maka ambil seluruh isi `$_POST` dan simpan ke dalam variabel. Kemudian kita sisipkan query `INSERT INTO` ke variabel `$sql` dengan menambahkan nama tabel user dan values seluruh data yang diinputkan user. Kemudian jika query tersebut menghasilkan nilai `true` maka data telah berhasil diinputkan, sebaliknya jika tidak maka data gagal diinputkan.

d. Tampilan dan penjelasan code read pada index.php

```

1 <!-- Start Table -->
2 <div class="container" data-aos="fade-up" data-aos-duration="2000">
3 <div class="page-header clearfix">
4 <a href="data_insert.php" class="btn btn-outline-primary mb-3">Tambah Data</a>
5 </div>
6 <table class="table table-bordered text-center">
7 <thead>
8 <tr>
9 <th scope="col">No Induk</th>
10 <th scope="col">Nama</th>
11 <th scope="col">Email</th>
12 <th scope="col">Tanggal Lahir</th>
13 <th scope="col">Tempat Lahir</th>
14 <th scope="col">Jenis Kelamin</th>
15 <th scope="col">Agama</th>
16 <th scope="col">Alamat</th>
17 <th scope="col">No Telepon</th>
18 <th scope="col">Kelas</th>
19 <th scope="col" colspan="2">Pengaturan</th>
20 </tr>
21 </thead>
22 <tbody>
23 <?php
24 $sql = "SELECT * FROM user";
25 $query = mysqli_query($db, $sql);
26
27 while ($siswa = mysqli_fetch_array($query)) {
28     echo <tr>;
29     echo <td> . $siswa['nomor_induk'] . "</td>";
30     echo <td> . $siswa['nama'] . "</td>";
31     echo <td> . $siswa['email'] . "</td>";
32     echo <td> . $siswa['birth_date'] . "</td>";
33     echo <td> . $siswa['birth_place'] . "</td>";
34     echo <td> . $siswa['gender'] . "</td>";
35     echo <td> . $siswa['religion'] . "</td>";
36     echo <td> . $siswa['address'] . "</td>";
37     echo <td> . $siswa['telepon'] . "</td>";
38     echo <td> . $siswa['kelas'] . "</td>";
39     echo <td><a class="badge badge-warning" href="data_update.php?nomor_induk=" . $siswa['nomor_induk'] . ">Edit</a></td>";
40     echo <td><a class="badge badge-danger" href="hapus.php?nomor_induk=" . $siswa['nomor_induk'] . ">Hapus</a></td>";
41     echo </tr>;
42 }
43 </tbody>
44 </table>
45
46 <p>Jumlah Siswa adalah <?php echo mysqli_num_rows($query) ?></p>

```

Penjelasan:

Seluruh tampilan memiliki navbar dan footer yang sama, oleh itu hanya content yang akan dijelaskan lebih detail. Dalam proses read kita akan mengambil seluruh data yang ada dengan query `SELECT * FROM user`, kemudian ditampilkan dalam bentuk tabel serta menambahkan fitur update / edit dan delete / hapus di paling kanan tabel. Index.php akan menampilkan seluruh data yang telah di create, update, atau delete.

- e. Tampilan dan penjelasan code update pada data_update.php dan proses_update.php

```
C: > xampp > htdocs > Praktikum 8 > data_update.php
1  <?php
2
3  include("config.php");
4
5  // kalau tidak ada id di query string
6  if (!isset($_GET['nomor_induk'])) {
7      header('Location: index.php');
8  }
9
10 //ambil id dari query string
11 $id = $_GET['nomor_induk'];
12
13 // buat query untuk ambil data dari database
14 $sql = "SELECT * FROM user WHERE nomor_induk=$id";
15 $query = mysqli_query($db, $sql);
16 $siswa = mysqli_fetch_assoc($query);
17
18 // jika data yang di-edit tidak ditemukan
19 if (mysqli_num_rows($query) < 1) {
20     die("data tidak ditemukan...");
21 }
22
```

```
<!-- Start form -->
<div class="container" data-aos="fade-up" data-aos-duration="2000">
  <form action="proses_update.php" method="POST">
    <div class="form-group">
      <label for="nomor_induk">*Nomor Induk</label>
      <input type="text" name="nomor_induk" class="form-control" id="nomor_induk" value="{php echo $siswa['nomor_induk']; }>" aria-describedby="hint" readonly>
      <small id="hint" class="form-text text-danger">Nomor Induk tidak dapat diubah.</small>
    </div>
    <div class="form-group">
      <label for="nama">*Nama</label>
      <input type="text" name="nama" class="form-control" id="nama" value="{php echo $siswa['nama']; }>" aria-describedby="hint" readonly>
      <small id="hint" class="form-text text-danger">Nama tidak dapat diubah.</small>
    </div>
    <div class="form-group">
      <label for="email">*Email</label>
      <input type="email" class="form-control" name="email" id="email" value="{php echo $siswa['email']; }>">
    </div>
    <div class="form-group">
      <label for="birth_date">*Tanggal Lahir</label>
      <input type="date" class="form-control" name="birth_date" id="birth_date" value="{php echo $siswa['birth_date']; }>">
    </div>
    <div class="form-group">
      <label for="birth_place">*Tempat Lahir</label>
      <input type="text" class="form-control" name="birth_place" id="birth_place" value="{php echo $siswa['birth_place']; }>">
    </div>
    <div class="form-group">
      <label for="religion">*Agama</label>
      <select class="form-control" id="religion" name="religion">
        <option value="Hindu" selected="selected">Hindu</option>
        <option value="Islam">Islam</option>
        <option value="Buddha">Buddha</option>
        <option value="Katolik">Katolik</option>
        <option value="Protestan">Protestan</option>
        <option value="Konghucu">Konghucu</option>
      </select>
    </div>
  </form>
</div>
```

Penjelasan:

Pertama – tama untuk update kita akan mengambil seluruh data dengan `SELECT * FROM user` dan kondisi `WHERE nomor_induk` nya sama dengan id yang di edit/update karena akan kita tampilkan pada kolom formulir sebagai data sebelum diedit. Nilai yang diambil sebelumnya ditampilkan pada value dari input formulir pada code di atas. Terdapat kondisi nilai seperti nomor_induk dan nama yang tidak dapat diubah dengan menggunakan property `readonly` pada html. User dapat mengklik tombol submit/simpan untuk mengubah data pada database.

```
C:\xampp\htdocs> Praktikum 8 > proses_update.php
1 <?php
2
3 include("config.php");
4
5 // cek apakah tombol simpan sudah diklik atau belum?
6 if (isset($_POST['update'])) {
7
8     // ambil data dari formulir
9     $nomor_induk = $_POST['nomor_induk'];
10    $name = $_POST['name'];
11    $email = $_POST['email'];
12    $birth_date = $_POST['birth_date'];
13    $birth_place = $_POST['birth_place'];
14    $gender = $_POST['gender'];
15    $religion = $_POST['religion'];
16    $address = $_POST['address'];
17    $telp = $_POST['telp'];
18    $class = $_POST['class'];
19
20    // buat query update
21    $sql = "UPDATE user SET nomor_induk='$nomor_induk', name='$name', email='$email', birth_date='$birth_date', birth_place='$birth_place', gender='$gender', religion='$religion',
22    address='$address', telp='$telp', class='$class' WHERE nomor_induk='$nomor_induk'";
23    $query = mysqli_query($db, $sql);
24
25    // apakah query update berhasil?
26    if ($query) {
27        // kalau berhasil alihkan ke halaman index.php
28        header('location: index.php');
29    } else {
30        // kalau gagal tampilkan pesan
31        die("Gagal menyimpan perubahan...");
32    }
33 } else {
34     die("Akses dilarang...");
35 }
```

Penjelasan:

Setelah user mengklik tombol submit, maka seluruh data yang diedit akan dimasukan ke database dengan query `UPDATE user SET` kolom dengan nilai yang diinputkan user. Jika berhasil maka langsung diarahkan ke `index.php`, dan sebaliknya akan muncul error.


f. Tampilan dan penjelasan code delete pada `hapus.php`

```
C:\xampp\htdocs> Praktikum 8 > hapus.php
1 <?php
2
3 include("config.php");
4
5 if (isset($_GET['nomor_induk'])) {
6
7     // ambil id dari query string
8     $nomor_induk = $_GET['nomor_induk'];
9
10    // buat query hapus
11    $sql = "DELETE FROM user WHERE nomor_induk=$nomor_induk";
12    $query = mysqli_query($db, $sql);
13
14    // apakah query hapus berhasil?
15    if ($query) {
16        header('location: index.php');
17    } else {
18        die("gagal menghapus...");
19    }
20 } else {
21     die("akses dilarang...");
22 }
```

Penjelasan:

Jika user mengklik tombol hapus pada tabel index.php, maka logika code di atas akan berjalan dengan mengambil nomor induk user dan kemudian menggunakan query DELETE FROM user dengan kondisi WHERE nomor_induk sama dengan nomor induk yang diklik tombol delete. Jika berhasil maka akan dikembalikan ke index.php dan jika tidak maka muncul error gagal menghapus.

g. Output CRUD

 Good Choice

HomeTambah Data

KELUAR

Tambah Data Sistem Informasi Good Choice

*Nomor Induk

*Nama

*Email

*Tanggal Lahir

*Tempat Lahir

*Jenis kelamin

*Agama

Gambar 1. Create

Selamat Datang Di Sistem Informasi Good Choice

[Tambah Data](#)

No Induk	Nama	Email	Tanggal Lahir	Tempat Lahir	Jenis Kelamin	Agama	Alamat	No Telepon	Kelas	Pengaturan
123	I Kadek Agus Chandra Pradika	guscan@gmail.com	2020-12-07	Medewi	Laki-laki	Hindu	Jimbaran	08123456789	B1	Edit Hapus
124	Theresia Girsang	theresia@gmail.com	2020-12-07	Medan	Perempuan	Katolik	Jimbaran	08123456789	B1	Edit Hapus
125	I Putu Sedana Wijaya	jaya@gmail.com	2020-12-09	Bali	Laki-laki	Buddha	Jimbaran	08123456789	C1	Edit Hapus
126	Yoga Mahendra	yogamahendra123@gmail.com	2000-02-02	Bali	Laki-laki	Buddha	Jimbaran Asri	0821222334567	PBW B	Edit Hapus

Jumlah Siswa adalah 4

Gambar 2. Read

Update Data Sistem Informasi Good Choice

*Nomor Induk

124

Nomor Induk tidak dapat diubah.

*Nama

Theresia Girsang

Nama tidak dapat diubah.

*Email

theresia@gmail.com

*Tanggal Lahir

12/07/2020

*Tempat Lahir

Medan

*Agama

Katolik

*Jenis kelamin

Perempuan

*No. Telp

08123456789

*Alamat

Jimbaran

*Kelas

B1

Gambar 3. Update

Selamat Datang Di Sistem Informasi Good Choice

[Tambah Data](#)

No Induk	Nama	Email	Tanggal Lahir	Tempat Lahir	Jenis Kelamin	Agama	Alamat	No Telepon	Kelas	Pengaturan
123	I Kadek Agus Chandra Pradika	guscan@gmail.com	2020-12-07	Medewi	Laki-laki	Hindu	Jimbaran	08123456789	B1	Edit Hapus
124	Theresia Girsang	theresia@gmail.com	2020-12-07	Medan	Perempuan	Katolik	Jimbaran	08123456789	B1	Edit Hapus
125	I Putu Sedana Wijaya	jaya@gmail.com	2020-12-09	Bali	Laki-laki	Buddha	Jimbaran	08123456789	C1	Edit Hapus
126	Yoga Mahendra	yogamahendra123@gmail.com	2000-02-02	Bali	Laki-laki	Buddha	Jimbaran Asri	0821222334455	PBW B	Edit Hapus

Jumlah Siswa adalah 4

Klik tombol delete berwarna merah di kanan table, maka data akan langsung terhapus pada tabel begitu juga pada databasenya.

No Induk	Nama	Email	Tanggal Lahir	Tempat Lahir	Jenis Kelamin	Agama	Alamat	No Telepon	Kelas	Pengaturan
123	I Kadek Agus Chandra Pradika	guscan@gmail.com	2020-12-07	Medewi	Laki-laki	Hindu	Jimbaran	08123456789	B1	Edit Hapus
124	Theresia Girsang	theresia@gmail.com	2020-12-07	Medan	Perempuan	Katolik	Jimbaran	08123456789	B1	Edit Hapus
125	I Putu Sedana Wijaya	jaya@gmail.com	2020-12-09	Bali	Laki-laki	Buddha	Jimbaran	08123456789	C1	Edit Hapus

Jumlah Siswa adalah 3

Gambar 4. Delete

BAB IV

PENUTUP

4.1 Simpulan

CRUD merupakan operasi yang digunakan untuk memanipulasi data database pada sebuah web. CRUD sebenarnya singkatan *Create, Read, Update, Delete*. *Create* digunakan untuk membuat atau menginput data, *Read* untuk menampilkan data, *Update* untuk mengedit atau mengubah data sedangkan *Delete* untuk menghapus data. Dengan CRUD kita bisa membuat form login, register, input, update dan hapus data dan masih banyak lagi. Pada pembuatan form selain CRUD kita juga wajjibkan menggunakan session agar nantinya web lebih aman karena dengan menggunakan session maka kita tidak bisa untuk langsung masuk ke halaman sesudah login karena setiap halaman akan dilakukan pengecekan session jadi apabila belum login maka halaman tidak bisa diakses.

4.2 Saran

Diharapkan dapat mengerti dan memahami konsep CRUD dengan baik. Mengetahui dan memahami penggunaan dan implementasi CRUD pada pembuatan website.

DAFTAR PUSTAKA

https://en.wikipedia.org/wiki/Create,_read,_update_and_delete . (Diakses pada 6 Desember 2020).

<https://www.sekolahkoding.com/perpustakaan/pengertian-crud-di-php-dan-mysql>. (Diakses pada 6 Desember 2020).