

LAPORAN PRAKTIKUM PEMROGRAMAN WEB



oleh:

FARKHAN 20081010060

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
JAWA TIMUR
2022**

1. JAVASCRIPT & JQUERY

1.1. SOAL dan JAWABAN

- 1) Buatlah sebuah tampilan segitiga bintang piramida yang mengarah ke empat sisi menggunakan javascript
- 2) Buatlah aplikasi web yang dapat melakukan proses CRUD menggunakan kombinasi JQuery dengan database yang dipilih dengan saudara adapun pilihannya Firebase,MySQL, dan WebSQL

2. PENULISAN KODE

KODE NOMOR 1

```
<div class="tab-header">
  <div class="pane active">Segitiga 1</div>
  <div class="pane">Segitiga 2</div>
  <div class="pane">Segitiga 3</div>
  <div class="pane">Segitiga 4</div>
</div>
```

Pada potongan kode diatas berfungsi sebagai tombol untuk berpindah-pindah form. Kelas “active” digunakan untuk memberikan style pada tombol yang sedang aktif.

```
<form id="segitiga1">
  <div class="txt_field">
    <input type="text" class="tinggi" required>
    <span></span>
    <label for="">Tinggi Segitiga</label>
  </div>
  <input type="button" value="Tampilkan" class="tampilkan"
onclick="segitiga1(this.form)">
</form>
```

Pada tag input terdapat properti onclik=”segitiga1(this.form)” yang berfungsi untuk memanggil fungsi segitiga1() pada file javascript serta this.form berarti mengirimkan *form* yang membungkus tak input tersebut sebagai parameter pada fungsi segitiga1. Setiap *form* akan memanggil fungsi yang berbeda yang telah diberi nama segitiga1, segitiga2, segitiga3, dan segitiga4 yang akan menampilkan bentuk segitiga yang berbeda-beda. Dapat dilihat tampilannya sebagai berikut.

```
<div class="tabs hasil">
  <h2>Hasil</h2>
  <p class="bintang"></p>
  <input type="button" value="Bersihkan" class="bersihkan" onclick="bersihkan()">
</div>
```

Tag <p> dengan kelas bintang digunakan sebagai wadah untuk menampilkan segitiga bintang. Kemudian di bawahnya terdapat tag <input> beserta properti onclick=”bersihkan” yang berfungsi untuk memanggil fungsi bersihkan(). Fungsi bersihkan dibuat untuk menghapus segitiga bintang setelah ditampilkan pada layar. Tampilannya dapat dilihat pada gambar di bawah.

File style.css :

```
body {
    background-color: #ddd;
    font-family: "Raleway", sans-serif;
}

.container {
    display: flex;
    padding-top: 5rem;
}

/* Tabs Style */
.tabs {
    position: absolute;
    width: 500px;
    height: 320px;
    background: #f5f5f5;
    padding: 20px 30px;
    overflow: hidden;
}

/* Button style */
input[type="button"] {
    width: 100%;
    height: 50px;
    border: 1px solid #00acee;
    background: #00acee;
    font-size: 18px;
    color: #f5f5f5;
    font-weight: 700;
    cursor: pointer;
    outline: none;
}
input[type="button"]:hover {
    border-color: #f5f5f5;
    transition: .5s;
}

/* Tabs Form Style */
.tabs.form .tab-header {
    height: 60px;
    display: flex;
    align-items: center;
}
.tabs.form .tab-header > div {
    width: calc(100% / 4);
    text-align: center;
    color: #888;
    font-weight: 600;
    cursor: pointer;
    font-size: 14px;
    text-transform: uppercase;
    outline: none;
}
.tabs.form .tab-header > div.active {
    color: #00acee;
}
.tabs.form .tab-indicator {
```

```

    position: relative;
    width: calc(100% / 4);
    height: 5px;
    background: #00acee;
    left: 0px;
    border-radius: 5px;
    transition: all 0.5s ease-in-out;
}
.tabs.form .tab-body {
    position: relative;
    height: calc(100% - 60px);
    padding: 10px 5px;
}
.tabs.form .tab-body > div {
    position: absolute;
    top: -200%;
    margin-top: 5px;
    transform: scale(0.9);
    transition: opacity 0.5s, ease-in-out 0s, transform 0.5s ease-in-out 0s;
}
.tabs.form .tab-body > div.active {
    top: 0%;
    opacity: 1;
    transform: scale(1);
    margin-top: 0px;
    width: 100%;
}
.tabs.form {
    right: 50.5%;
    box-shadow: -5px 10px 5px #ccc;
    border-radius: 50px 5px 5px 50px;
}

/* Tabs Hasil Style */
.tabs.hasil {
    left: 50.5%;
    box-shadow: 5px 10px 5px #ccc;
    border-radius: 5px 50px 50px 5px;
    padding-right: 40px;
    padding-left: 40px;
}
.hasil h2 {
    text-align: center;
    letter-spacing: 10px;
    text-transform: uppercase;
    font-weight: 900;
    color: black;
}
p.bintang {
    text-align: left;
    overflow-y: scroll;
    height: 55%;
    letter-spacing: normal;
    font-family: 'Courier New', Courier, monospace;
    color: black;
}
.bersihkan {
    padding: 0 40px;
    border-radius: 5px 5px 25px 5px;
}

```

```

}

/* Form Style */
.tabs.form h2 {
    text-align: center;
    color: black;
}
.tabs form {
    padding: 0 20px;
}
form .txt_field {
    position: relative;
    border-bottom: 2px solid #adadad;
    margin: 40px 0;
}
.txt_field input {
    width: 100%;
    padding: 0 5px;
    height: 40px;
    font-size: 16px;
    border: none;
    background: none;
    outline: none;
    color: black;
}
.txt_field label {
    position: absolute;
    top: 50%;
    left: 5px;
    color: #adadad;
    transform: translateY(-50%);
    font-size: 16px;
    pointer-events: none;
    transition: 0.5s;
}
.txt_field span::before {
    content: '';
    position: absolute;
    left: 0;
    width: 0;
    height: 2px;
    background: #00acee;
    transition: 0.5s;
}
.txt_field input:focus ~ label,
.txt_field input:valid ~ label {
    top: -5px;
    color: #00acee;
}
.txt_field input:focus ~ span::before,
.txt_field input:valid ~ span::before {
    width: 100%;
}
.tampilkan {
    border-radius: 5px 5px 5px 25px;
}

```

File script.js

```

let tabHeader = document.getElementsByClassName('tab-header')[0];
let tabIndicator = document.getElementsByClassName('tab-indicator')[0];
let tabBody = document.getElementsByClassName('tab-body')[0];

let tabsPane = tabHeader.getElementsByClassName('pane');

for(let i=0; i<tabsPane.length; i++) {
  tabsPane[i].addEventListener('click', function() {
    tabHeader.getElementsByClassName('active')[0].classList.remove('active');
    tabsPane[i].classList.add('active');

    tabBody.getElementsByClassName('active')[0].classList.remove('active');
    tabBody.getElementsByClassName('content')[i].classList.add('active');

    tabIndicator.style.left = `calc(calc(100% / 4) * ${i})`;
  });
}

```

Kode di atas digunakan untuk memindahkan berpindah pada form yang berbeda. Serta membantu mengatur style yang ada.

```

function segitiga1(form) {
  let tinggi = form.getElementsByClassName('tinggi')[0].value;
  let hasil = '';
  for (let i = 0; i < tinggi; i++) {
    for (let j = 0; j <= i; j++) {
      hasil += '* ';
    }
    hasil += '<br>';
  }
  let hasilArea = document.getElementsByClassName('bintang')[0];
  hasilArea.innerHTML = hasil;
}

let tinggi = form.getElementsByClassName('tinggi')[0].value;
let hasil = '';
for (let i = 0; i < tinggi; i++) {
  for (let j = tinggi; j > i; j--) {
    hasil += '*';
  }
  hasil += "<br>";
}
let hasilArea = document.getElementsByClassName('bintang')[0];
hasilArea.innerHTML = hasil;
}

function segitiga3(form) {
  let tinggi = form.getElementsByClassName('tinggi')[0].value;
  let hasil = '';
  for (let i = tinggi; i > 0; i--) {
    for (let j = 1; j <= tinggi; j++) {
      if (j >= i) {
        hasil += '*';
      } else {
        hasil += '&nbsp;';
      }
    }
    hasil += '<br>';
  }
}

```

```

    let hasilArea = document.getElementsByClassName('bintang')[0];
    hasilArea.innerHTML = hasil;
}

function segitiga4(form) {
    let tinggi = form.getElementsByClassName('tinggi')[0].value;
    let hasil = '';
    for (let i = tinggi; i > 0; i--) {
        for (let j = tinggi; j > 0; j--) {
            if (j > i) {
                hasil += '&nbsp;';
            } else {
                hasil += '*';
            }
        }
        hasil += '<br>';
    }
    let hasilArea = document.getElementsByClassName('bintang')[0];
    hasilArea.innerHTML = hasil;
}

function bersihkan() {
    let bintang = document.getElementsByClassName('bintang')[0];

```

Kode di atas digunakan untuk menampilkan hasil segitiga serta membersihkan hasil segitiga yang telah ditampilkan.

KODE NOMOR 2

database.js

```

module.exports = {
    multipleStatements: true,
    host: 'localhost',
    user: 'root',
    password: 'root',
    database: 'pemweb_sipinjam'
};

```

Pada potongan kode diatas berfungsi sebagai konfigurasi untuk menghubungkan ke database.

indeks.js

```

// library yang digunakan
const path = require('path');
const express = require('express');
const session = require('express-session');
const hbs = require('hbs');
const bodyParser = require('body-parser');
const flash = require('req-flash');
const app = express();

// Definisi lokasi file router
const bukuRoutes = require('./src/routes/router-buku');

// Konfigurasi library session
app.use(session({

```

```

    resave: false,
    saveUninitialized: false,
    secret: 't@1k0ch3ng',
    name: 'secretName',
    cookie: {
      sameSite: true
    },
  })),

//set views file
app.set('views',path.join(__dirname,'src/views'));
app.set('view engine', 'hbs');

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(flash());
//set folder public sebagai static folder untuk static file
app.use('/assets',express.static(__dirname + '/public'));

app.use(function(req, res, next) {
  res.setHeader('Cache-Control', 'no-cache, private, no-store, must-revalidate,
max-stale=0, post-check=0, pre-check=0');
  res.setHeader('Pragma', 'no-cache');
  next();
});

// route
app.use('/buku', bukuRoutes);

//server listening
app.listen(8000, () => {
  console.log('Server is running at port 8000');
});

```

Pada potongan kode diatas berfungsi untuk mengontrol semua alur pada website

Conntroller-buku.js

```

const config = require('../configs/database');
let mysql = require('mysql');
let pool = mysql.createPool(config);

pool.on('error',(err)=> {
  console.error(err);
});

module.exports = {
  index(req,res){
    res.render("buku",{
      url : 'http://localhost:8000/',
    });
  },
  all(req,res){
    let sql = "SELECT * FROM buku ORDER BY id_buku DESC";
    let query = pool.query(sql, (err, results) => {
      if(err) throw err;
      return res.json(results);
    });
  },
}

```



```

    save(req,res){
      let data = {judul: req.body.judul,
                  id_kategori: req.body.id_kategori,
                  penerbit: req.body.penerbit,
                  penulis: req.body.penulis,
                  tahun_terbit: req.body.tahun_terbit,
                  stok: req.body.stok};

      let sqlKategori = `UPDATE kategori SET jumlah = jumlah + 1 WHERE
id_kategori = ${req.body.id_kategori}`;
      let queryKategori = pool.query(sqlKategori, (err, results) => {
        if(err) throw err;
      });
      let sql = "INSERT INTO buku SET ?";
      let query = pool.query(sql, data,(err, results) => {
        if(err) throw err;
        res.redirect('/buku');
      });
    },
    delete(req,res){
      let sql = "DELETE FROM buku WHERE id_buku = "+req.body.id_buku;
      let query = pool.query(sql, req.params.id,(err, results) => {
        if(err) throw err;
        res.redirect('/buku');
      });
    },
    update(req,res){
      let sqlKategoriBefore = `UPDATE kategori SET
                              jumlah = jumlah - 1 WHERE id_kategori =
${req.body.id_kategori_before}`;
      let queryKategoriBefore = pool.query(sqlKategoriBefore, (err, results) => {
        if(err) throw err;
      });

      let sqlKategoriAfter = `UPDATE kategori SET
                              jumlah = jumlah + 1 WHERE id_kategori =
${req.body.id_kategori_after}`;
      let queryKategoriAfter = pool.query(sqlKategoriAfter, (err, results) => {
        if(err) throw err;
      });

      let sql = "UPDATE buku SET judul='"+req.body.judul
                +"', id_kategori='"+req.body.id_kategori_after
                +"', penerbit='"+req.body.penerbit
                +"', penulis='"+req.body.penulis
                +"', tahun_terbit='"+req.body.tahun_terbit
                +"', stok='"+req.body.stok
                +"' WHERE id_buku='"+req.body.id_buku;
      let query = pool.query(sql, (err, results) => {
        if(err) throw err;
        res.redirect('/buku');
      });
    }
  }
}

```

Pada potongan kode diatas berfungsi sebagai controller yang mengatur proses CRUD data buku pada website.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Buku | Sipinjam</title>

  <!-- Material CDN -->
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Sharp"
rel="stylesheet">
  <!-- My CSS -->
  <link rel="stylesheet" href="/assets/css/style.css">
</head>
<body>
  <div class="container">
    <!-- ASIDE -->
    <aside>
      <div class="top">
        <div class="logo">
          
          <h2>SI<span class="danger">PINJAM</span></h2>
        </div>
        <div class="close" id="close-btn">
          <span class="material-icons-sharp">close</span>
        </div>
      </div>

      <div class="sidebar">
        <a href="/dashboard">
          <span class="material-icons-
sharp">grid_view</span>
          <h3>Dashboard</h3>
        </a>
        <a href="/peminjaman">
          <span class="material-icons-
sharp">receipt_long</span>
          <h3>Peminjaman</h3>
        </a>
        <a href="/pengembalian">
          <span class="material-icons-sharp">receipt</span>
          <h3>Pengembalian</h3>
        </a>
        <a href="/buku" class="active">
          <span class="material-icons-
sharp">library_books</span>
          <h3>Buku</h3>
        </a>
        <a href="/anggota">
          <span class="material-icons-sharp">groups</span>
          <h3>Anggota</h3>
        </a>
        <a href="/petugas">
          <span class="material-icons-
sharp">person_outline</span>
          <h3>Petugas</h3>
        </a>
      </div>
    </aside>
  </div>

```

```

        <a href="/login/logout">
            <span class="material-icons-sharp">logout</span>
            <h3>Logout</h3>
        </a>
    </div>
</aside>
<!-- END OF ASIDE -->

<!-- MAIN CONTENT -->
<main>
    <h1>Buku</h1>

    <div class="card-form">
        <h2>Tambah Buku</h2>
        <div class="form">
            <form action="/buku/save" method="post">
                <div class="form-control">
                    <label for="judul">Judul
Buku</label>
                    <input type="text" name="judul"
id="judul" placeholder="Judul buku">
                </div>
                <div class="form-control">
                    <label for="id_kategori">Kategori
Buku</label>
                    <select name="id_kategori"
id="id_kategori">

                        </select>
                </div>
                <div class="form-control">
                    <label
for="penerbit">Penerbit</label>
                    <input type="text" name="penerbit"
id="penerbit" placeholder="Penerbit buku">
                </div>
                <div class="form-control">
                    <label for="penulis">Penulis</label>
                    <input type="text" name="penulis"
id="penulis" placeholder="Penulis buku">
                </div>
                <div class="form-control">
                    <label for="tahun_terbit">Tahun
Terbit</label>
                    <input type="number"
name="tahun_terbit" id="tahun_terbit" placeholder="Tahun terbit buku">
                </div>
                <div class="form-control">
                    <label for="stok">Stok</label>
                    <input type="number" name="stok"
id="stok" placeholder="Ketersediaan buku">
                </div>
                <div class="submit">
                    <button
type="submit">Tambah</button>
                </div>
            </form>
        </div>
    </div>
</main>

```

```

        <div class="recent-orders">
            <h2>Daftar Buku</h2>
            <table id="tabel-buku">

                </table>
                <!-- PAGINATION -->
                <nav aria-label="Page navigation example">
                    <ul class="pagination justify-content-center"
id="pagination-buku">

                        </ul>
                    </nav>
                    <a href="#" onclick="generatePDF()">Download PDF</a>
                </div>

                <div class="recent-orders">
                    <h2>Bar Chart Kategori Buku</h2>
                    <canvas id="bukuBarChart"></canvas>
                    <a href=""></a>
                </div>
            </main>
            <!-- END OF MAIN -->

            <div class="right">
                <div class="top">
                    <button id="menu-btn">
                        <span class="material-icons-sharp">menu</span>
                    </button>
                    <div class="theme-toggler">
                        <span class="material-icons-sharp
active">light_mode</span>
                        <span class="material-icons-
sharp">dark_mode</span>
                    </div>
                    <div class="profile">
                        <div class="info">
                            <p>Hey, <b id="nama-petugas"></b></p>
                            <small class="text-muted">Petugas</small>
                        </div>
                        <div class="profile-photo">
                            
                        </div>
                    </div>
                </div>
                <!-- END OF TOP -->

                <div class="sales-analytics" id="kategori-buku">
                    <h2>Kategori Buku</h2>
                    <div class="item add-kategori">
                        <div>
                            <span class="material-icons-
sharp">add</span>
                            <h3>Add Kategori</h3>
                        </div>
                    </div>
                </div>

                <div class="recent-updates">

```

```

                <h2>Chart Kategori Buku</h2>
                <div class="updates">
                    <canvas id="chartKategori"></canvas>
                </div>
            </div>
        <!-- END OF CHART KATEGORI -->
    </div>
</div>

{{!-- Modal Edit --}}
<div class="modal-container" id="editModal">
    <div class="modal">
        <h2>Ubah Data Anggota</h2>
        <div class="form">
            <form action="/buku/update" method="post">
                <input type="hidden" class="id" name="id_buku">
                <div class="form-control">
                    <label for="judul">Judul Buku</label>
                    <input type="text" name="judul"
class="judul">
                </div>
                <input type="hidden" name="id_kategori_before"
id="id_kategori_before">
                <div class="form-control">
                    <label for="id_kategori">Kategori
Buku</label>
                    <select name="id_kategori_after"
class="id_kategori_after">

                        </select>
                    </div>
                <div class="form-control">
                    <label for="penerbit">Penerbit</label>
                    <input type="text" name="penerbit"
class="penerbit">
                </div>
                <div class="form-control">
                    <label for="penulis">Penulis</label>
                    <input type="text" name="penulis"
class="penulis">
                </div>
                <div class="form-control">
                    <label for="tahun_terbit">Tahun
Terbit</label>
                    <input type="number" name="tahun_terbit"
class="tahun_terbit">
                </div>
                <div class="form-control">
                    <label for="stok">Stok</label>
                    <input type="number" name="stok"
class="stok">
                </div>
                <div class="submit">
                    <button type="submit">Ubah</button>
                    <a type="button"
id="closeEditModal">Batal</a>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
    {{!-- End Modal Edit --}}

    {{!-- Modal Hapus --}}
    <div class="modal-container" id="hapusModal">
        <div class="modal">
            <h2>Hapus Data Buku</h2>
            <div class="form">
                <form action="/buku/delete" method="post">
                    <input type="hidden" class="id_buku"
name="id_buku">

                    <p>Anda yakin mau menghapus data ini?</p>
                    <div class="submit">
                        <button type="submit">Ya!</button>
                        <a type="button"
id="closeHapusModal">Batal</a>
                    </div>
                </form>
            </div>
        </div>
    </div>
    {{!-- End Modal Hapus --}}

    {{!-- Modal Add Kategori --}}
    <div class="modal-container" id="addKategoriModal">
        <div class="modal">
            <h2>Tambah Kategori Buku</h2>
            <div class="form">
                <form action="/kategori/save" method="post">
                    <div class="form-control">
                        <label for="kategori">Kategori Buku</label>
                        <input type="text" name="kategori"
class="kategori">

                    </div>
                    <div class="submit">
                        <button type="submit">Tambah</button>
                        <a type="button"
id="closeAddKategoriModal">Batal</a>
                    </div>
                </form>
            </div>
        </div>
    </div>
    {{!-- End Add Kategori Modal --}}

    {{!-- Modal Hapus Kategori --}}
    <div class="modal-container" id="hapusKategoriModal">
        <div class="modal">
            <h2>Hapus Kategori</h2>
            <div class="form">
                <form action="/kategori/delete" method="post">
                    <input type="hidden" class="id_kategori"
name="id_kategori">

                    <p>Anda yakin mau menghapus data ini?</p>
                    <div class="submit">
                        <button type="submit">Ya!</button>
                        <a type="button"
id="closeHapusKategoriModal">Batal</a>
                    </div>
                </form>
            </div>
        </div>
    </div>
    {{!-- End Modal Hapus Kategori --}}

```

```

                                </div>
                            </form>
                        </div>
                    </div>
                </div>
                {{!-- End Modal Hapus Kategori --}}

                {{!-- JQuery --}}
                <script src="/assets/js/jquery-3.3.1.js"></script>

                {{!-- Chart JS --}}
                <script
src="https://cdn.jsdelivr.net/npm/chart.js@3.7.1/dist/chart.min.js"></script>

                {{!-- pdf --}}
                <script
src="https://cdn.jsdelivr.net/npm/jspdf@2.5.1/jspdf.umd.min.js"></script
>

                {{!-- My Script --}}
                <script src="/assets/js/script.js"></script>
                <script src="/assets/js/functions.js"></script>

                {{!-- Page Script --}}
                <script src="/assets/js/buku.js"></script>
</body>
</html>

```

Pada potongan kode diatas berfungsi untuk membuat tampilan pada website.

Router-buku.js

```

const router = require('express').Router();
const bukuController = require('../controllers/controller-buku');
const verifyUser = require('../configs/verify');

router.get('/', verifyUser.isLogin, bukuController.index);
router.get('/all', verifyUser.isLogin, bukuController.all);
router.post('/save', verifyUser.isLogin, bukuController.save);
router.post('/delete', verifyUser.isLogin, bukuController.delete);
router.post('/update', verifyUser.isLogin, bukuController.update);

module.exports = router;

```

Pada potongan kode diatas berisi daftar route yang mengarahkan ke link dari setiap proses CRUD.

buku.js

```

$(document).ready(function() {
    getBuku.loadData = "/buku/all";
    getKategori.loadData = "/kategori/all";
    getPetugas.loadData = "/petugas/profile";

    kategoriChart();
});

window.jsPDF = window.jspdf.jsPDF;

```

```

const getBuku = {
  set loadData(data) {
    const url = "http://localhost:8000" + data;
    Functions.prototype.getRequest(getBuku, url);
  },
  set successData(response) {
    paginatedResult(response, $("#tabel-buku"), $("#pagination-buku"), 1);
  }
}

const getKategori = {
  set loadData(data) {
    const url = "http://localhost:8000" + data;
    Functions.prototype.getRequest(getKategori, url);
  },
  set successData(response) {
    let kategori = response;
    const allKategori = $(".sales-analytics");
    for (let i = 0; i < kategori.length; i++) {
      allKategori.append(`
        <div class="item online">
          <div class="right">
            <div class="info">
              <h3 class="nama-kategori">${kategori[i].kategori}</h3>
            </div>
            <h3 class="danger delete" data-
id="${kategori[i].id_kategori}">Hapus</h3>
            </div>
          </div>
        `);
    }

    // for form add
    const option = document.createElement("option");
    option.value = "";
    option.innerHTML = "Pilih Kategori";
    document.getElementById("id_kategori").appendChild(option);
    for (let i = 0; i < kategori.length; i++) {
      const option = document.createElement("option");
      option.value = kategori[i].id_kategori;
      option.innerHTML = kategori[i].kategori;
      document.getElementById("id_kategori").appendChild(option);
    }

    // for form update
    const optionUpdate = document.createElement("option");
    optionUpdate.value = "";
    optionUpdate.innerHTML = "Pilih Kategori";

    document.getElementsByClassName("id_kategori")[0].appendChild(optionUpdate);
    for (let i = 0; i < kategori.length; i++) {
      const optionUpdate = document.createElement("option");
      optionUpdate.value = kategori[i].id_kategori;
      optionUpdate.innerHTML = kategori[i].kategori;

      document.getElementsByClassName("id_kategori_after")[0].appendChild(optionUpdate);
    }
  }
}

```



```

const editModal = document.querySelector('#editModal');
const closeEditModal = document.getElementById('closeEditModal');
const hapusModal = document.querySelector('#hapusModal');
const closeHapusModal = document.getElementById('closeHapusModal');

const addKategoriModal = document.querySelector('#addKategoriModal');
const addKategori = document.querySelector('.add-kategori');
const closeAddKategoriModal = document.getElementById('closeAddKategoriModal');
const hapusKategoriModal = document.querySelector('#hapusKategoriModal');
const closeHapusKategoriModal =
document.getElementById('closeHapusKategoriModal');

$('#tabel-buku').on('click', '.edit', function() {
    var id = $(this).data('id');
    var judul = $(this).data('judul');
    var kategori = $(this).data('kategori');
    var penerbit = $(this).data('penerbit');
    var penulis = $(this).data('penulis');
    var tahun_terbit = $(this).data('tahun_terbit');
    var stok = $(this).data('stok');
    $('#id').val(id);
    $('#judul').val(judul);
    $('#id_kategori_before').val(kategori);
    $('#id_kategori_after').val(kategori);
    $('#penerbit').val(penerbit);
    $('#penulis').val(penulis);
    $('#tahun_terbit').val(tahun_terbit);
    $('#stok').val(stok);

    editModal.classList.add('show');
})
closeEditModal.addEventListener('click', function() {
    editModal.classList.remove('show');
});

$('#tabel-buku').on('click', '.delete', function() {
    var id_buku = $(this).data('id');
    $('#id_buku').val(id_buku);
    hapusModal.classList.add('show');
})
closeHapusModal.addEventListener('click', function() {
    hapusModal.classList.remove('show');
});

addKategori.addEventListener('click', function() {
    addKategoriModal.classList.add('show');
});
closeAddKategoriModal.addEventListener('click', function() {
    addKategoriModal.classList.remove('show');
});

$('#kategori-buku').on('click', '.delete', function() {
    var id_kategori = $(this).data('id');
    $('#id_kategori').val(id_kategori);
    hapusKategoriModal.classList.add('show');
});
closeHapusKategoriModal.addEventListener('click', function() {

```

```

        hapusKategoriModal.classList.remove('show');
    });

    function generatePDF() {
        reportBuku.loadData = "/buku/all";
    }

    const reportBuku = {
        set loadData(data) {
            const url = "http://localhost:8000" + data;
            Functions.prototype.getRequest(reportBuku, url);
        },
        set successData(response) {
            var buku = "Daftar Buku Sipinjam \n";
            for (let i = 0; i < response.length; i++) {
                let judul = JSON.stringify(response[i].judul);
                let penulis = JSON.stringify(response[i].penulis);
                let penerbit = JSON.stringify(response[i].penerbit);
                let tahun_terbit = JSON.stringify(response[i].tahun_terbit);
                let stok = JSON.stringify(response[i].stok);
                buku = buku + "\n"
                    + "Judul : " + judul + "\n"
                    + "Penulis : " + penulis + "\n"
                    + "Penerbit : " + penerbit + "\n"
                    + "Tahun Terbit : " + tahun_terbit + "\n"
                    + "Stok : " + stok + "\n";
            }
            const doc = new jsPDF();
            doc.text(buku, 10, 10);
            doc.save("Laporan Buku.pdf");
        }
    }

    const getPetugas = {
        set loadData(data) {
            const url = "http://localhost:8000" + data;
            Functions.prototype.getRequest(getPetugas, url);
        },
        set successData(response) {
            const data = response;
            document.getElementById("nama-petugas").innerHTML = data[0].nama;
        }
    }

    // pie chart kategori
    function kategoriChart() {
        async function fetchData() {
            const url = "http://localhost:8000/kategori/all";
            const response = await fetch(url);
            const data = await response.json();
            return data;
        }

        fetchData().then(data => {
            const kategori = data.map(
                function(index) {return index.kategori}
            );
            const jumlah = data.map(
                function(index) {return index.jumlah}
            );
        });
    }

```

```

    });

    myKategoriChart.config.data.labels = kategori;
    myKategoriChart.config.data.datasets[0].data = jumlah;
    myKategoriChart.update();

    // bar kategori char
    myBarKategoriChart.config.data.labels = kategori;
    myBarKategoriChart.config.data.datasets[0].data = jumlah;
    myBarKategoriChart.update();
  });
}

const dataKategori = {
  datasets: [{
    label: 'Kategori Buku',
    backgroundColor: [
      'rgba(255, 99, 132, 1)',
      'rgba(54, 162, 235, 1)',
      'rgba(255, 206, 86, 1)',
      'rgba(75, 192, 192, 1)',
      'rgba(153, 102, 255, 1)',
      'rgba(255, 159, 64, 1)'
    ],borderWidth: 1
  }]
}

const configKategori = {
  type: 'pie',
  data: dataKategori,
}

const myKategoriChart = new Chart(
  document.getElementById('chartKategori'),
  configKategori
);
// end pie chart kategori

// bar chart kategori
const configBarKategori = {
  type: 'bar',
  data: dataKategori,
}

const myBarKategoriChart = new Chart(
  document.getElementById('bukuBarChart'),
  configBarKategori
);
// end bar chart kategori

// pagination
function paginatedResult(response, table, pagination, page) {
  const limit = 10

  const startIndex = (page - 1) * limit
  const endIndex = page * limit

  const results = {}

```

```

    if (endIndex < response.length) {
        results.next = {
            page: page + 1,
        }
    }
    if (startIndex > 0) {
        results.previous = {
            page: page - 1,
        }
    }

    results.results = response.slice(startIndex, endIndex)

    table.empty();
    table.append(`
        <tr>
            <th width="35%">Judul</th>
            <th>Pengarang</th>
            <th>Penerbit</th>
            <th>Tahun</th>
            <th>Stok</th>
            <th></th>
        </tr>
    `);
    for (let i = 0; i < results.results.length; i++) {
        table.append(`
            <tr>
                <td>${results.results[i].judul}</td>
                <td>${results.results[i].penulis}</td>
                <td>${results.results[i].penerbit}</td>
                <td>${results.results[i].tahun_terbit}</td>
                <td>${results.results[i].stok}</td>
                <td class="danger delete" data-
id="${results.results[i].id_buku}">Hapus</td>
                <td class="primary edit" data-id="${results.results[i].id_buku}"
                    data-judul="${results.results[i].judul}"
                    data-
kategori="${results.results[i].id_kategori}"
                    data-
penulis="${results.results[i].penulis}"
                    data-
penerbit="${results.results[i].penerbit}"
                    data-
tahun_terbit="${results.results[i].tahun_terbit}"
                    data-
stok="${results.results[i].stok}">Edit</td>
            </tr>
        `);
    }

    pagination.empty();
    if (results.previous) {
        pagination.append(`
            <li class="page-item">
                <a class="page-link" href="#" data-page="${results.previous.page}"
aria-label="Previous">
                    <span aria-hidden="true">&laquo;</span>
                    <span class="sr-only">Previous</span>
                </a>

```

```

        </li>
    `);
} else {
    pagination.append(`
        <li class="page-item disabled">
            <a class="page-link" href="#" aria-label="Previous">
                <span aria-hidden="true">&laquo;</span>
                <span class="sr-only">Previous</span>
            </a>
        </li>
    `);
}

for (let i = 1; i <= Math.ceil(response.length / limit); i++) {
    if (i == page) {
        pagination.append(`
            <li class="page-item active">
                <a class="page-link" href="#" data-page="${i}">${i}</a>
            </li>
        `);
    } else {
        pagination.append(`
            <li class="page-item">
                <a class="page-link" href="#" data-page="${i}">${i}</a>
            </li>
        `);
    }
}

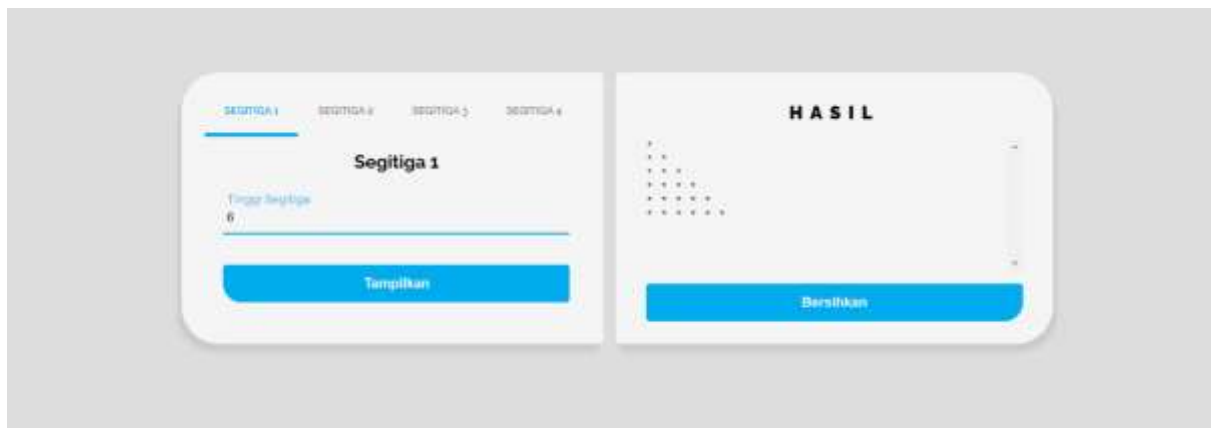
if (results.next) {
    pagination.append(`
        <li class="page-item">
            <a class="page-link" href="#" data-page="${results.next.page}"
aria-label="Next">
                <span aria-hidden="true">&raquo;</span>
                <span class="sr-only">Next</span>
            </a>
        </li>
    `);
} else {
    pagination.append(`
        <li class="page-item disabled">
            <a class="page-link" href="#" aria-label="Next">
                <span aria-hidden="true">&raquo;</span>
                <span class="sr-only">Next</span>
            </a>
        </li>
    `);
}

$('.page-link').click(function(e) {
    e.preventDefault();
    const page = $(this).data('page');
    paginatedResult(response, table, pagination, page);
});
}

```

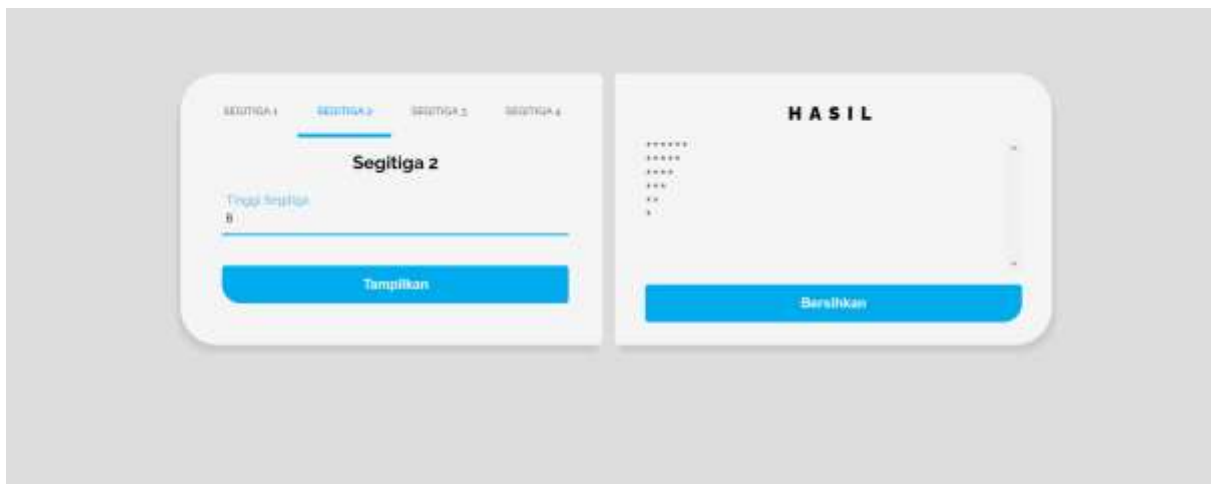
Pada potongan kode diatas berfungsi untuk menghubungkan file tampilan website dengan proses CRUD.

3. SCREENSHOT HASIL PERCOBAAN



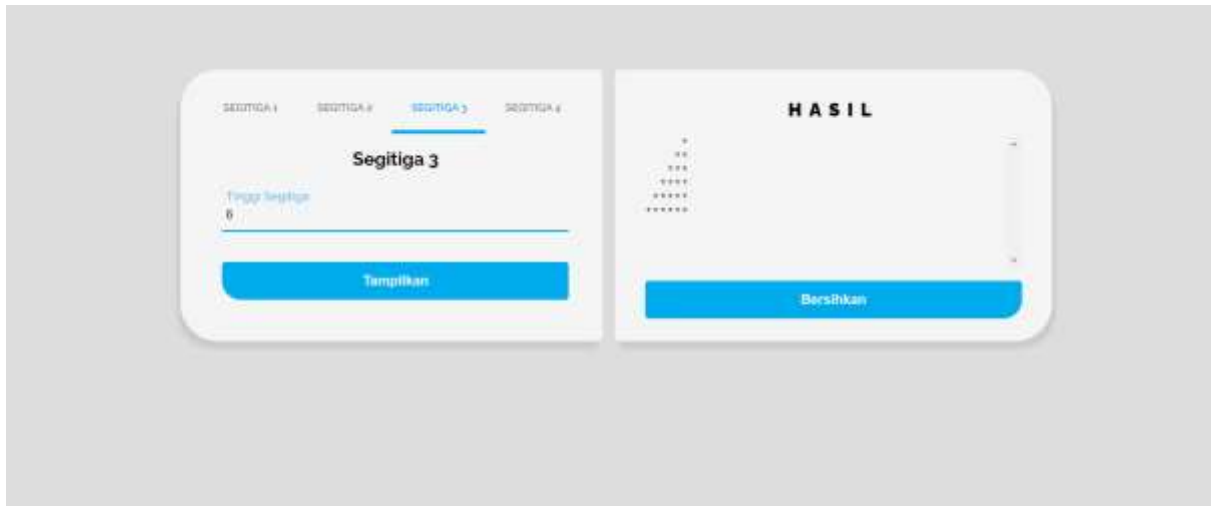
Gambar 3.1. Segitiga siku-siku pertama

Pada gambar 3.1. menunjukkan hasil segitiga siku-siku pertama



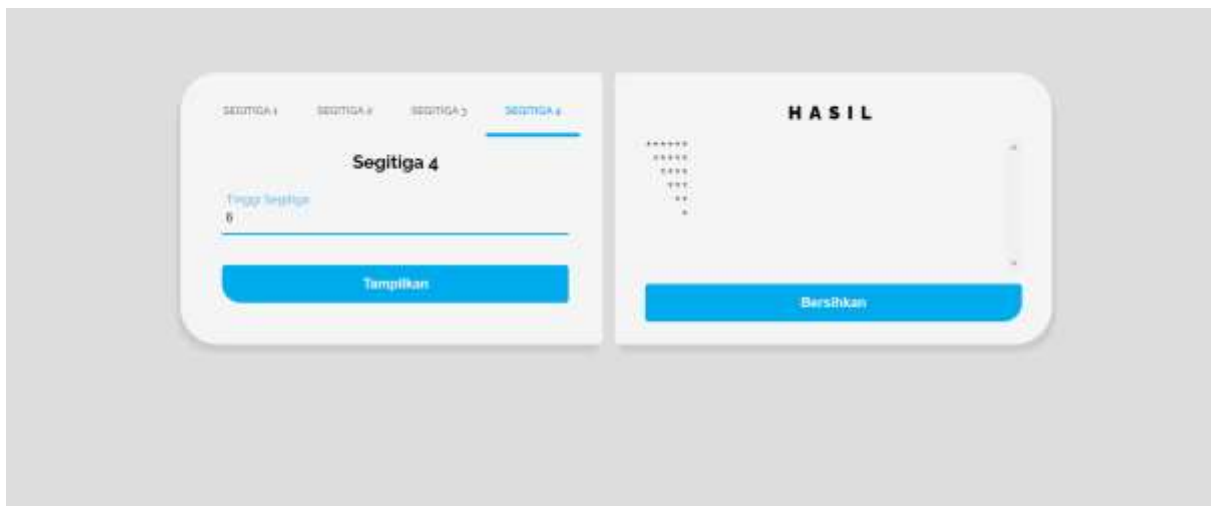
Gambar 3.2. Segitiga siku-siku kedua

Pada gambar 3.2. menunjukkan hasil segitiga siku-siku kedua



Gambar 3.3. Segitiga siku-siku ketiga

Pada gambar 3.3. menunjukkan hasil segitiga siku-siku ketiga



Gambar 3.4. Segitiga siku-siku keempat

Pada gambar 3.4. menunjukkan hasil segitiga siku-siku keempat

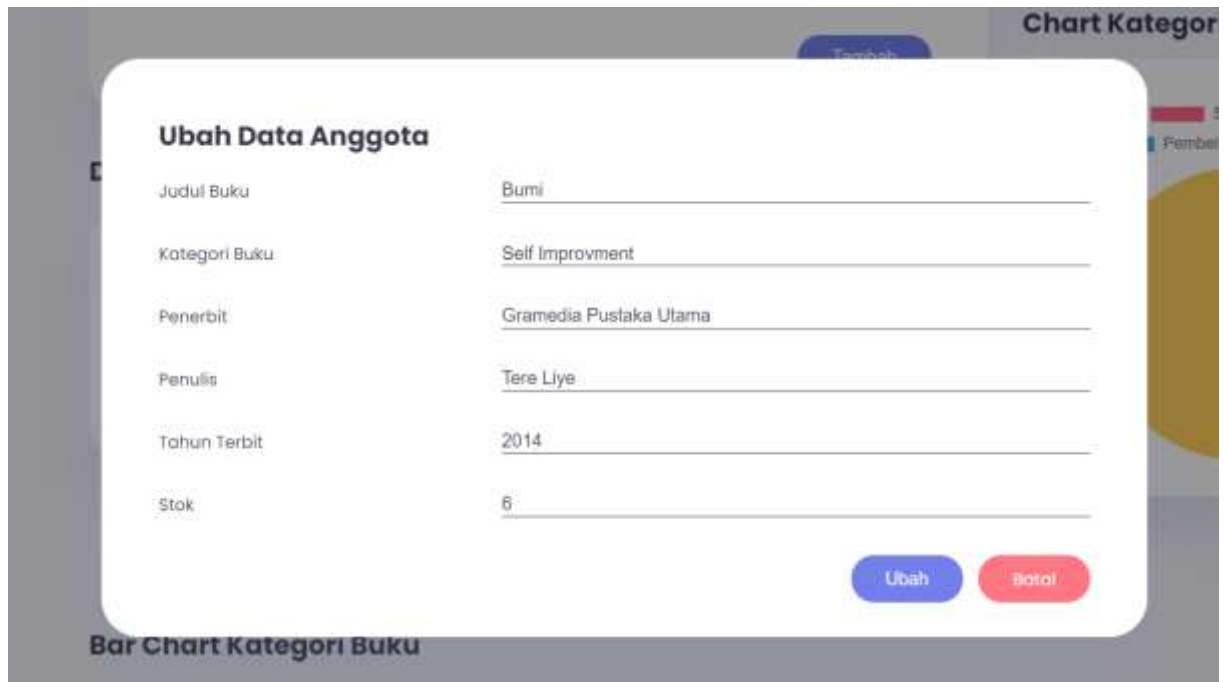
Gambar 3.5. Tambah Buku

Pada gambar 3.5. menunjukkan tampilan dari halaman tambah buku

Judul	Pengarang	Penerbit	Tahun	Stok	
Bumi	Tere Liye	Gramedia Pustaka Utama	2014	6	Hapus Edit
Bumi Manusia	Pramoedya Ananta Toer	penerbit 2	2013	9	Hapus Edit
5 cm	Donny Dhirgantoro	penerbit 1	2012	10	Hapus Edit
Cedera Kepala	M. Z. Arifin	Sagung Seto	2013	5	Hapus Edit

Gambar 3.6. Daftar Buku

Pada gambar 3.6. menunjukkan tampilan dari daftar buku



Ubah Data Anggota

Judul Buku	Bumi
Kategori Buku	Self Improvement
Penerbit	Gramedia Pustaka Utama
Penulis	Tere Liye
Tahun Terbit	2014
Stok	6

Ubah Batal

Gambar 3.7. Ubah Data Buku

Pada gambar 3.7. menunjukkan tampilan form ubah data buku