

DAFTAR PUSTAKA

- Coppock, M. (2021). The best web browsers for 2021. Diakses Juni, 2021, from Digital Trends website: <https://www.digitaltrends.com/computing/best-browser-chrome-vs-firefox-vs-safari-vs-edge/>
- Dexie.js - Minimalistic IndexedDB Wrapper. (2014). Diakses Juni, 2021, from Dexie.org website: <https://dexie.org/>
- Indexed Database API 2.0. (2018). Diakses Juni, 2021, from W3.org website: <https://www.w3.org/TR/IndexedDB-2/>
- IndexedDB | Can I use... Support tables for HTML5, CSS3, etc. (2021). Diakses Juni, 2021, from Caniuse.com website: <https://caniuse.com/?search=indexedDB>
- IndexedDB API - Web APIs | MDN. (2021). Diakses Juni, 2021, from Mozilla.org website: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API
- IndexedDB. (2021). IndexedDB. Diakses Juni, 2021, from Javascript.info website: <https://javascript.info/indexeddb>
- Marshall, C. (2021). The best web browsers in 2021. Diakses Juni, 2021, from TechRadar website: <https://www.techradar.com/best/browser>
- Muchmore, M. (2021). Chrome, Edge, Firefox, Opera, or Safari: Which Browser Is Best? Diakses Juni, 2021, from PCMAG website: <https://www.pcmag.com/picks/chrome-edge-firefox-opera-or-safari-which-browser-is-best>
- Arefipour, S., & Mozahhebi, M. (2015). Comparison of IndexedDB and SQLite Based on Developers' Concerns.
- Belloro, S., & Mylonas, A. (2018). Security considerations around the usage of client-side storage APIs.
- Camden, R. (2015). *Client-side data storage: keeping it local*. " O'Reilly Media, Inc."

- Chandra, E., & Kistijantoro, A. I. (2017, August). Database development supporting offline update using CRDT:(Conflict-free replicated data types). In *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)* (pp. 1-6). IEEE.
- Janik, A., & Kiebzak, S. (2014). Client-side Storage. Offline Availability Of Data. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 3-10.
- Kimak, S., & Ellman, J. (2013). Performance testing and comparison of client side databases versus server side. *Northumbria University*.
- Kimak, S., & Ellman, J. (2015). HTML5 IndexedDB Encryption: Prevention against Potential Attacks. *International Journal of Intelligent Computing Research*, 6(4), 621-630.
- Kimak, S., & Ellman, J. (2015, December). The role of HTML5 IndexedDB, the past, present and future. In *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)* (pp. 379-383). IEEE.
- Laine, M. (2012). Client-side storage in web applications. *Aalto University*.
- MacDonald, M. (2013). *HTML5: The missing manual*. " O'Reilly Media, Inc."
- Maennel, O. M. A NOVEL APPROACH FOR SECURING HTML5 CLIENT-SIDE DATABASE, INDEXEDDB.
- Marshall, S. (2013). *Creating offline web applications using HTML5: a thesis presented in partial fulfillment of the requirements for the degree of Master of Information Sciences in Computer Science,[Massey University, Albany, New Zealand]* (Doctoral dissertation, Massey University).
- Sleit, A. (2017, May). Evaluating IndexedDB performance on web browsers. In *2017 8th International Conference on Information Technology (ICIT)* (pp. 488-494). IEEE.
- Tahir, Z., Inamoto, T., Higami, Y., & Kobayashi, S. (2015, November). The Analysis of Automated HTML5 Offline Services (AHOS). In *2015 International Conference on Intelligent Informatics and BioMedical*

Sciences (ICIIBMS) (pp. 62-66). IEEE.

Top 10 BEST Browsers For PC [2021 Browser Ranking]. (2021). Diakses Juni, 2021, from Softwaretestinghelp.com website:
<https://www.softwaretestinghelp.com/best-browser-ranking/>

LAMPIRAN

A. Source Code

- index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <title>Book Store</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous" />
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css"
    integrity="sha384-50oBUHEmvpQ+11W4y57PTFmhCaXp0ML5d60M1M7uH2+nqUivzIebhnd0JK28anvf" crossorigin="anonymous" />
  <link rel="stylesheet" href="./style.css" />
</head>

<body>
  <main>
    <div class="container text-center">
      <h1 class="bg-light py-4 text-info">
        <i class="fas fa-swatchbook"></i> Book Store
      </h1>
      <div class="d-flex justify-content-center">
        <form class="w-50">
          <input type="text" id="userid" class="form-control" autocomplete="off" readonly placeholder="ID" />
          <input type="text" id="praname" class="form-control" autocomplete="off" placeholder="Product Name" />
          <div class="row">
            <div class="col">
              <input type="text" id="seller" class="form-control m-0" autocomplete="off" placeholder="Seller" />
            </div>
            <div class="col">
              <input type="text" id="price" class="form-control m-0" autocomplete="off" placeholder="Price" />
            </div>
          </div>
        </form>
      </div>

      <div class="d-flex justify-content-center">
        <button class="btn btn-success" id="btn-create">Create</button>
        <button class="btn btn-primary" id="btn-read">Read</button>
        <button class="btn btn-warning" id="btn-update">Update</button>
        <button class="btn btn-danger" id="btn-delete">Delete All</button>
      </div>

      <!-- Table -->
      <div class="d-flex table-data">
        <table class="table table-striped">
          <thead>
            <tr>
              <th scope="col">ID</th>
              <th scope="col">Product Name</th>
              <th scope="col">Seller</th>
              <th scope="col">Price</th>
              <th scope="col">Edit</th>
              <th scope="col">Delete</th>
            </tr>
          </thead>
          <tbody id="tbody">
            <!--
              <tr>
                <th scope="row">1</th>
                <td>Mark</td>
                <td>Otto</td>
                <td>@mdo</td>
              </tr>
              <tr>
                <th scope="row">2</th>
                <td>Jacob</td>
                <td>Thornton</td>
                <td>@fat</td>
              </tr>
            -->
          </tbody>
        </table>
      </div>
    </main>
  </body>
</html>
```

```

                <th scope="row">2</th>
                <td>Jacob</td>
                <td>Thornton</td>
                <td>@fat</td>
            </tr>
        -->
    </tbody>
</table>
</div>
<div id="notfound"></div>
</div>
</main>

<!-- Labels -->
<div class="w-100 btn btn-success insertmsg">Data Inserted Successfully...!</div>
<div class="w-100 btn btn-warning updatemsg">Data Updated..! Refresh Database To See Result</div>
<div class="w-100 btn btn-danger deletemsg">Data Deleted..!</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
    integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous">
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
    integrity="sha384-U02eT0CpHqQdSjQ6hJty5KVphtPhzWj9WO1c1HTMga3DDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous">
</script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
    integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEeFFn/nJGzIxFDs-f4x0xIM+B07jRM" crossorigin="anonymous">
</script>

<!-- Include Dexie -->
<script src="https://unpkg.com/dexie@latest/dist/dexie.js"></script>

<!-- Custom main file-->
<script src="../js/main.js" type="module"></script>
</body>
</html>

```

- main.js

```

import prodb, {
  bulkcreate,
  createEle,
  getData,
  SortObj
} from "./module.js";

let db = prodb("Productdb", {
  products: `++id, name, seller, price`
});

// input tags
const userid = document.getElementById("userid");
const proname = document.getElementById("praname");
const seller = document.getElementById("seller");
const price = document.getElementById("price");

// create button
const btncreate = document.getElementById("btn-create");
const btnread = document.getElementById("btn-read");
const btnupdate = document.getElementById("btn-update");
const btndelete = document.getElementById("btn-delete");

// user data

// event listener for create button
btncreate.onclick = event => {
  // insert values
  let flag = bulkcreate(db.products, {
    name: proname.value,
    seller: seller.value,
    price: price.value
  });
  // reset textbox values
  //praname.value = "";

```

```

//seller.value = "";
// price.value = "";
proname.value = seller.value = price.value = "";

// set id textbox value
getData(db.products, data => {
  userid.value = data.id + 1 || 1;
});
table();

let insertmsg = document.querySelector(".insertmsg");
getMsg(flag, insertmsg);
};

// event listener for create button
btnread.onclick = table;

// button update
btnupdate.onclick = () => {
  const id = parseInt(userid.value || 0);
  if (id) {
    // call dexie update method
    db.products.update(id, {
      name: proname.value,
      seller: seller.value,
      price: price.value
    }).then((updated) => {
      // let get = updated ? `data updated` : `couldn't update data`;
      let get = updated ? true : false;

      // display message
      let updatemsg = document.querySelector(".updatemsg");
      getMsg(get, updatemsg);

      proname.value = seller.value = price.value = "";
      //console.log(get);
    })
  } else {
    console.log(`Please Select id: ${id}`);
  }
}

// delete button
btndelete.onclick = () => {
  db.delete();
  db = proddb("Productdb", {
    products: `++id, name, seller, price`
  });
  db.open();
  table();
  textID(userid);
  // display message
  let deletemsg = document.querySelector(".deletemsg");
  getMsg(true, deletemsg);
}

window.onload = event => {
  // set id textbox value
  textID(userid);
};

// create dynamic table
function table() {
  const tbody = document.getElementById("tbody");
  const notfound = document.getElementById("notfound");
  notfound.textContent = "";
  // remove all childs from the dom first
  while (tbody.hasChildNodes()) {
    tbody.removeChild(tbody.firstChild);
  }
}

```

```

}

getData(db.products, (data, index) => {
  if (data) {
    createEle("tr", tbody, tr => {
      for (const value in data) {
        createEle("td", tr, td => {
          td.textContent = data.price === data[value] ? `$$$ ${data[value]}` : data[value];
        });
      }
      createEle("td", tr, td => {
        createEle("i", td, i => {
          i.className += "fas fa-edit btnedit";
          i.setAttribute("data-id", data.id);
          // store number of edit buttons
          i.onclick = editbtn;
        });
      });
    });
  }
  createEle("td", tr, td => {
    createEle("i", td, i => {
      i.className += "fas fa-trash-alt btndelete";
      i.setAttribute("data-id", data.id);
      // store number of edit buttons
      i.onclick = deletebtn;
    });
  });
});
} else {
  notfound.textContent = "No record found in the database...!";
}

});
}

const editbtn = (event) => {

  let id = parseInt(event.target.dataset.id);
  db.products.get(id, function (data) {
    let newdata = SortObj(data);
    userid.value = newdata.id || 0;
    proname.value = newdata.name || "";
    seller.value = newdata.seller || "";
    price.value = newdata.price || "";
  });
}

// delete icon remove element
const deletebtn = event => {
  let id = parseInt(event.target.dataset.id);
  db.products.delete(id);
  table();
}

// textbox id
function textID(textboxid) {
  getData(db.products, data => {
    textboxid.value = data.id + 1 || 1;
  });
}

// function msg
function getMsg(flag, element) {
  if (flag) {
    // call msg
    element.className += " movedown";

    setTimeout(() => {
      element.classList.forEach(classname => {
        classname == "movedown" ? undefined : element.classList.remove('movedown');
      });
    }, 4000);
  }
}
}

```

- module.js

```

const productsdb = (dbname, table) => {
  const db = new Dexie(dbname);
  db.version(1).stores(table);
  db.open();

  return db;
  /**
   * const db = new Dexie('myDb');
   * db.version(1).stores({
   *   friends: `name, age`
   * });
   */
};

const bulkcreate = (dbtable, data) => {
  let flag = empty(data);
  if (flag) {
    dbtable.bulkAdd([data]);
    console.log("data inserted successfully...!");
  } else {
    console.log("Please provide data...!");
  }
  return flag;
};

// create dynamic elements
const createEle = (tagname, appendTo, fn) => {
  const element = document.createElement(tagname);
  if (appendTo) appendTo.appendChild(element);
  if (fn) fn(element);
};

// check textbox validation
const empty = object => {
  let flag = false;
  for (const value in object) {
    if (object[value] != "" && object.hasOwnProperty(value)) {
      flag = true;
    } else {
      flag = false;
    }
  }
  return flag;
};

// getData from the database
const getData = (dbname, fn) => {
  let index = 0;
  let obj = {};
  dbname.count(count => {
    // count rows in the table using count method
    if (count) {
      dbname.each(table => {
        // table => return the table object data
        // to arrange order we are going to create for in loop
        obj = SortObj(table);
        fn(obj, index++); // call function with data argument
      });
    } else {
      fn(obj);
    }
  });
};

const SortObj = (sortobj) => {
  let obj = {};
  obj = {
    id: sortobj.id,
    name: sortobj.name,
    seller: sortobj.seller,
    price: sortobj.price
  };
  return obj;
}

export default productsdb;
export {
  bulkcreate,
  createEle,
  getData,
  SortObj
};

```

LEMBAR PERBAIKAN SKRIPSI

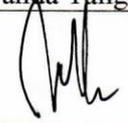
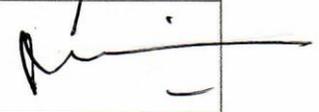
“STUDI DAN ANALISIS KINERJA INDEXEDDB PADA CLIENT WEB BROWSER”

OLEH:

**WINDA ASTIYANTI AZIS
D421 14 015**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 6 Agustus 2021.
Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari
penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Eng. Zulkifli Tahir, S.T., M.Sc.	
Sekretaris	A. Ais Prayogi Alimuddin, S.T., M.Eng.	
Anggota	Dr. Amil Ahmad Ilham, S.T., M.IT.	
	Dr. Eng. Muhammad Niswar, S.T., M.IT.	
	Dr. Indrabayu, S.T., M.T., M.Bus.Sys.	

Persetujuan perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Eng. Zulkifli Tahir, S.T., M.Sc.	
II	A. Ais Prayogi Alimuddin, S.T., M.Eng.	