

**WEB-BASED STUDENT BIODATA MANAGEMENT APPLICATION USING
HTML, CSS, JAVASCRIPT, AND LOCALSTORAGE**

**APLIKASI MANAJEMEN BIODATA MAHASISWA BERBASIS WEB
MENGUNAKAN HTML, CSS, JAVASCRIPT, DAN LOCALSTORAGE**

**Muh. Arfah Wahlil Pratama¹, Yusni², Adelia Asmarani³, Listi Aulia⁴, Annisa⁵,
Kharismatul Khomsiah⁶, Serli⁷**

Universitas Muhammadiyah Kolaka Utara, Lasusua, Indonesia ^{1,2,3,4,5,6,7}

muharfahwahlilpratama@gmail.com¹, yusnisist956@gmail.com², adeliaaa130726@gmail.com³,

listiauliaaa@gmail.com⁴, an3704605@gmail.com⁵, kharismatul3@gmail.com⁶,

seri.kolut13@gmail.com⁷

ABSTRACT

This research aims to design and implement a web-based student biodata management application equipped with a login system and CRUD (Create, Read, Update, Delete) features. Manual student biodata management still has several limitations, such as inefficient data recording, risk of data loss, and difficulty in updating information. Therefore, a web-based application is proposed as a simple solution to improve the effectiveness of student data management. The application was developed using HTML to structure the pages, CSS to design an interface resembling a laptop screen, and JavaScript to process the application logic. Data storage was implemented using the browser's LocalStorage, allowing the application to run without a server or external database. The login system functions as an access control mechanism before users enter the main biodata management page. The testing results show that the application can perform login authentication, input student biodata, save data, display data in a table, and edit or delete data effectively. Data stored in LocalStorage remain available even when the web page is reloaded or the browser is closed. In addition, the simple and structured interface makes the application easy to operate. Based on these results, this web-based student biodata management application can be used as a simple tool for managing student data and as a learning medium for understanding basic web application development, especially the integration of user interface design, programming logic, and client-side data storage.

Keywords: Web Application, Student Biodata, CRUD, JavaScript, LocalStorage.

ABSTRAK

Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi manajemen biodata mahasiswa berbasis web yang dilengkapi dengan sistem login dan fitur CRUD (*Create, Read, Update, Delete*). Pengelolaan biodata mahasiswa secara manual masih memiliki beberapa keterbatasan, seperti proses pencatatan yang kurang efisien, risiko kehilangan data, dan kesulitan dalam pembaruan informasi. Oleh karena itu, aplikasi berbasis web dikembangkan sebagai solusi sederhana untuk meningkatkan efektivitas pengelolaan data mahasiswa. Aplikasi ini dibuat menggunakan HTML untuk membentuk struktur halaman, CSS untuk mengatur tampilan antarmuka menyerupai layar laptop, serta JavaScript untuk mengolah logika aplikasi. Penyimpanan data dilakukan menggunakan LocalStorage pada browser, sehingga aplikasi dapat berjalan tanpa server maupun basis data eksternal. Sistem login diterapkan sebagai mekanisme pembatas akses sebelum pengguna masuk ke halaman utama pengelolaan biodata. Hasil pengujian menunjukkan bahwa aplikasi dapat menjalankan fungsi login, menambahkan biodata mahasiswa, menyimpan data, menampilkan data dalam tabel, serta melakukan pengeditan dan penghapusan data dengan baik. Data yang tersimpan melalui LocalStorage tetap tersedia meskipun halaman dimuat ulang atau browser ditutup. Selain itu, antarmuka yang sederhana dan terstruktur memudahkan pengguna dalam mengoperasikan aplikasi. Berdasarkan hasil tersebut, aplikasi manajemen biodata mahasiswa berbasis web ini dapat digunakan sebagai solusi sederhana dalam pengelolaan data mahasiswa serta sebagai media pembelajaran untuk memahami konsep dasar pengembangan aplikasi web, khususnya integrasi antara antarmuka pengguna, logika pemrograman, dan penyimpanan data di sisi klien.

Kata Kunci: Aplikasi Web, Biodata Mahasiswa, CRUD, JavaScript, LocalStorage.

This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Artikel ini adalah artikel akses terbuka yang didistribusikan di bawah ketentuan Lisensi Creative Commons Attribution 4.0 International (CC BY 4.0).



PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi dalam lima tahun terakhir (2020–2025) mengalami peningkatan yang sangat pesat, terutama pada pemanfaatan teknologi berbasis web di berbagai sektor, termasuk sektor pendidikan. Transformasi digital yang semakin masif mendorong institusi pendidikan untuk beradaptasi dengan sistem informasi yang lebih efektif, efisien, dan mudah diakses. Pemanfaatan aplikasi berbasis web menjadi salah satu solusi yang banyak digunakan karena dapat diakses secara fleksibel melalui berbagai perangkat tanpa memerlukan instalasi khusus.

Di lingkungan perguruan tinggi, pengelolaan data mahasiswa merupakan salah satu aspek penting dalam mendukung kegiatan akademik dan administrasi. Biodata mahasiswa mencakup informasi dasar yang digunakan dalam berbagai proses, seperti administrasi perkuliahan, pelaporan akademik, dan layanan kemahasiswaan. Namun, pada praktiknya, pengelolaan biodata masih sering dilakukan secara manual atau semi-digital, sehingga berpotensi menimbulkan permasalahan seperti keterlambatan pengolahan data, kesalahan pencatatan, dan risiko kehilangan data.

Seiring dengan perkembangan teknologi web, penggunaan bahasa pemrograman seperti HTML, CSS, dan JavaScript semakin luas karena kemudahan implementasinya. Teknologi tersebut memungkinkan pengembangan aplikasi web sederhana namun fungsional tanpa harus menggunakan server atau basis data yang kompleks. Salah satu fitur yang banyak dimanfaatkan dalam pengembangan aplikasi web skala kecil adalah `LocalStorage`, yang memungkinkan penyimpanan data langsung pada browser pengguna.

Pada periode 2020–2025, tren pengembangan aplikasi web juga mengarah pada penerapan sistem yang user-friendly dan interaktif. Aplikasi tidak hanya dituntut mampu menyimpan dan menampilkan data, tetapi juga menyediakan fitur pengelolaan data seperti penambahan, pengeditan, dan penghapusan data atau yang dikenal dengan konsep CRUD (*Create, Read, Update, Delete*). Selain itu, penerapan sistem login menjadi salah satu fitur penting untuk membatasi akses pengguna dan meningkatkan keamanan data.

Meskipun berbagai penelitian telah membahas pengembangan sistem informasi berbasis web untuk pengelolaan data mahasiswa, sebagian besar menggunakan arsitektur client-server dengan basis data seperti MySQL atau PostgreSQL yang memerlukan infrastruktur server dan biaya operasional (Maliki & Solichin, 2020; Sari et al., 2022). Sementara itu, solusi berbasis `LocalStorage` yang memanfaatkan penyimpanan di sisi klien masih jarang dieksplorasi secara mendalam, terutama untuk keperluan pembelajaran dan pengembangan prototipe sederhana tanpa ketergantungan server.

Penelitian ini memberikan kontribusi berupa implementasi sistem manajemen biodata mahasiswa berbasis web yang sederhana namun fungsional dengan memanfaatkan teknologi *client-side storage* (`LocalStorage`) tanpa memerlukan server atau database eksternal. Aplikasi ini dapat menjadi media pembelajaran bagi pengembang pemula untuk memahami konsep dasar pengembangan aplikasi web, khususnya integrasi antara antarmuka pengguna, logika pemrograman JavaScript, dan penyimpanan data di sisi klien. Selain itu, penelitian ini juga menghadirkan studi kasus penggunaan `LocalStorage` sebagai alternatif penyimpanan data untuk aplikasi web skala kecil yang tidak memerlukan infrastruktur server yang kompleks.

METODE PENELITIAN

Metode penelitian yang digunakan dalam pengembangan aplikasi manajemen biodata mahasiswa berbasis web ini adalah metode rekayasa perangkat lunak (software engineering) dengan pendekatan waterfall yang telah dimodifikasi. Metode ini bertujuan untuk merancang, membangun, dan menguji aplikasi yang mampu mengelola data biodata secara terstruktur, efektif, dan mudah digunakan. Proses penelitian terdiri dari lima tahapan utama, yaitu: (1) analisis kebutuhan sistem, (2) perancangan sistem, (3) implementasi sistem, (4) pengujian sistem, dan (5) evaluasi hasil pengujian.

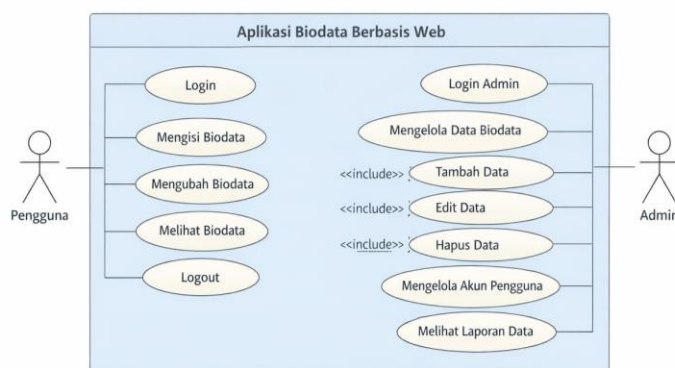
Tahap 1: Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem dilakukan untuk mengidentifikasi kebutuhan fungsional dan nonfungsional aplikasi. Kebutuhan fungsional meliputi fitur-fitur yang harus tersedia dalam aplikasi, sedangkan kebutuhan nonfungsional berkaitan dengan aspek kinerja, keamanan, dan kemudahan penggunaan aplikasi.

Kebutuhan Fungsional: 1. Sistem login untuk autentikasi pengguna sebelum mengakses halaman utama 2. Fitur input data biodata mahasiswa (nama lengkap, NIM, jurusan, jenis kelamin, alamat) 3. Fitur penyimpanan data menggunakan LocalStorage 4. Fitur penampilan data dalam bentuk tabel 5. Fitur pengeditan data biodata mahasiswa 6. Fitur penghapusan data biodata mahasiswa 7. Fitur reset form input

Kebutuhan Nonfungsional: 1. Aplikasi dapat dijalankan pada browser modern (Chrome, Firefox, Edge, Safari) 2. Antarmuka pengguna sederhana dan responsif 3. Data tetap tersimpan meskipun browser ditutup atau halaman dimuat ulang 4. Tidak memerlukan koneksi internet setelah aplikasi dimuat pertama kali 5. Waktu respons sistem kurang dari 1 detik untuk setiap operasi

Hasil analisis kebutuhan divisualisasikan menggunakan *Use Case Diagram* untuk menggambarkan interaksi antara pengguna (admin) dan sistem secara jelas. Use Case Diagram menunjukkan bahwa pengguna dapat melakukan login, mengelola data biodata (tambah, ubah, hapus), menampilkan data, dan logout dari sistem.

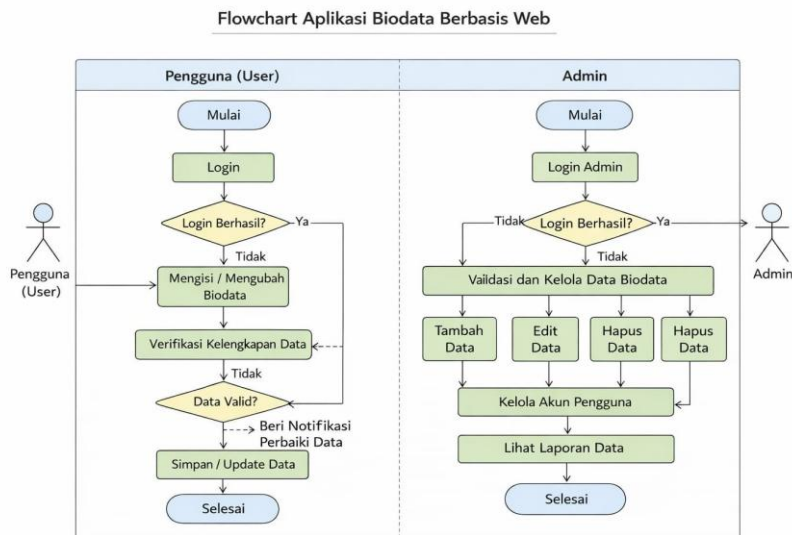


Gambar 1. Use Case Diagram

Tahap 2: Perancangan Sistem

Tahap perancangan system bertujuan untuk menggambarkan alur kerja aplikasi secara sistematis menggunakan *flowchart*. *Flowchart* dirancang untuk menunjukkan proses mulai dari pengguna melakukan login, system melakukan validasi username dan password, pengelolaan data biodata (tambah, ubah, hapus, tampil), hingga proses logout. *Flowchart* digunakan untuk memastikan alur system berjalan logis dan mudah dipahami sebelum tahap implementasi dilakukan.

Alur sistem dimulai ketika pengguna membuka aplikasi dan diarahkan ke halaman login. Sistem kemudian memvalidasi kredensial yang dimasukkan. Jika valid, pengguna diarahkan ke halaman manajemen biodata. Jika tidak valid, sistem menampilkan pesan error dan pengguna diminta untuk memasukkan kredensial kembali. Pada halaman manajemen biodata, pengguna dapat melakukan berbagai operasi CRUD. Setiap operasi akan memicu proses penyimpanan atau pengambilan data dari LocalStorage browser



Gambar 2. Flowchart Sistem

Tahap 3: Implementasi Sistem

Tahap implementasi sistem dilakukan dengan membangun aplikasi berbasis web menggunakan teknologi front-end, yaitu HTML5 untuk struktur halaman, CSS3 untuk styling dan layout, serta JavaScript (ES6) untuk logika aplikasi dan manipulasi DOM. Penyimpanan data memanfaatkan LocalStorage API yang tersedia pada browser modern.

Teknologi yang Digunakan: - **HTML5:** Digunakan untuk membentuk struktur halaman web, termasuk form input, tabel data, dan elemen-elemen antarmuka lainnya - **CSS3:** Digunakan untuk mengatur tampilan visual aplikasi, termasuk layout, warna, font, dan responsivitas - **JavaScript (ES6):** Digunakan untuk mengimplementasikan logika aplikasi, validasi input, manipulasi DOM, dan interaksi dengan LocalStorage - **LocalStorage API:** Digunakan sebagai mekanisme penyimpanan data di sisi klien dengan kapasitas penyimpanan sekitar 5-10 MB per domain

Aplikasi dikembangkan untuk dapat berjalan melalui browser tanpa memerlukan instalasi tambahan atau koneksi ke *server backend*, sehingga memudahkan pengguna dalam mengakses dan mengelola data biodata secara offline setelah aplikasi dimuat pertama kali.

Struktur Penyimpanan Data: Data mahasiswa disimpan dalam format JSON pada LocalStorage dengan struktur sebagai berikut:

```

[
  {
    "id": "unique_id",
    "nama": "Nama Lengkap",
    "nim": "123456",
    "jurusan": "Teknik Informatika",
    "jenisKelamin": "Laki-laki",
    "alamat": "Alamat Lengkap"
  }
]
  
```

Tahap 4: Pengujian Sistem

Tahap pengujian sistem dilakukan menggunakan metode *black box testing*, yang bertujuan untuk memastikan setiap fungsi aplikasi berjalan sesuai dengan kebutuhan yang telah

ditentukan. Pengujian dilakukan dengan cara menguji setiap fitur aplikasi tanpa memperhatikan struktur internal kode program.

Tabel 1. Skenario Pengujian BlackBox Testing

| No | Fitur yang Diuji | Input | Output yang Diharapkan | Hasil |
|----|----------------------------------------|-------------------------------------------------|---------------------------------------------------|------------|
| 1 | Login dengan kredensial benar | Username: admin, Password: admin123 | Berhasil login dan diarahkan ke halaman utama | Berhasil ✓ |
| 2 | Login dengan kredensial salah | Username: admin, Password: salah | Muncul pesan error "Username atau password salah" | Berhasil ✓ |
| 3 | Tambah data biodata mahasiswa | Nama, NIM, Jurusan, Jenis Kelamin, Alamat | Data tersimpan dan muncul di tabel | Berhasil ✓ |
| 4 | Validasi form kosong | Submit form tanpa mengisi data | Muncul pesan validasi "Harap isi semua field" | Berhasil ✓ |
| 5 | Edit data biodata mahasiswa | Ubah data nama mahasiswa | Data terupdate di tabel dan LocalStorage | Berhasil ✓ |
| 6 | Hapus data biodata mahasiswa | Klik tombol hapus pada baris data | Data terhapus dari tabel dan LocalStorage | Berhasil ✓ |
| 7 | Reset form | Klik tombol reset setelah mengisi form | Semua field form kembali kosong | Berhasil ✓ |
| 8 | Persistensi data setelah reload | Reload halaman browser | Data tetap tampil di tabel | Berhasil ✓ |
| 9 | Persistensi data setelah tutup browser | Tutup dan buka kembali browser | Data tetap tersimpan dan dapat diakses | Berhasil ✓ |

Hasil pengujian menunjukkan bahwa semua fitur aplikasi berfungsi dengan baik sesuai dengan spesifikasi yang telah ditentukan.

Tahap 5: Evaluasi Hasil Pengujian

Berdasarkan hasil pengujian black box testing, dapat diidentifikasi beberapa kelebihan dan keterbatasan sistem:

Kelebihan: 1. Aplikasi dapat berjalan tanpa memerlukan server atau database eksternal 2. Proses CRUD berjalan dengan cepat dan responsif 3. Data tetap tersimpan meskipun browser ditutup atau halaman dimuat ulang 4. Antarmuka sederhana dan mudah digunakan 5. Cocok untuk pembelajaran konsep dasar pengembangan web front-end

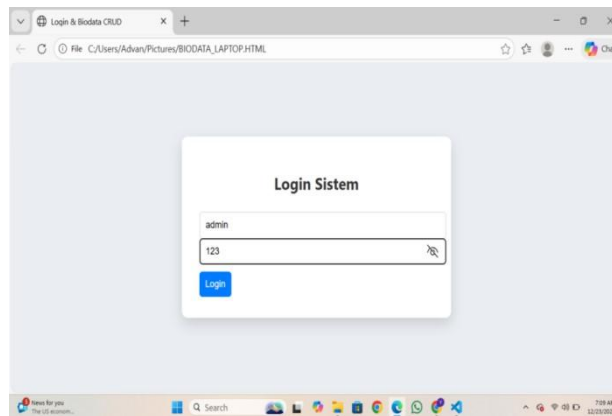
Keterbatasan: 1. Data hanya tersimpan di browser lokal, tidak dapat diakses dari perangkat lain 2. Kapasitas penyimpanan terbatas (sekitar 5-10 MB per domain) 3. Data dapat hilang jika pengguna menghapus cache atau data browser 4. Tidak ada enkripsi data, sehingga kurang aman untuk data sensitif 5. Tidak mendukung sistem multi-user secara bersamaan 6. Tidak cocok untuk aplikasi skala besar dengan jumlah data yang sangat banyak

HASIL DAN PEMBAHASAN

Hasil Implementasi Aplikasi

Aplikasi manajemen biodata mahasiswa berbasis web telah berhasil dikembangkan dengan memanfaatkan teknologi HTML, CSS, JavaScript, dan LocalStorage. Aplikasi ini terdiri dari dua halaman utama, yaitu halaman login dan halaman manajemen biodata mahasiswa. Berikut adalah hasil implementasi dari setiap halaman beserta fungsinya.

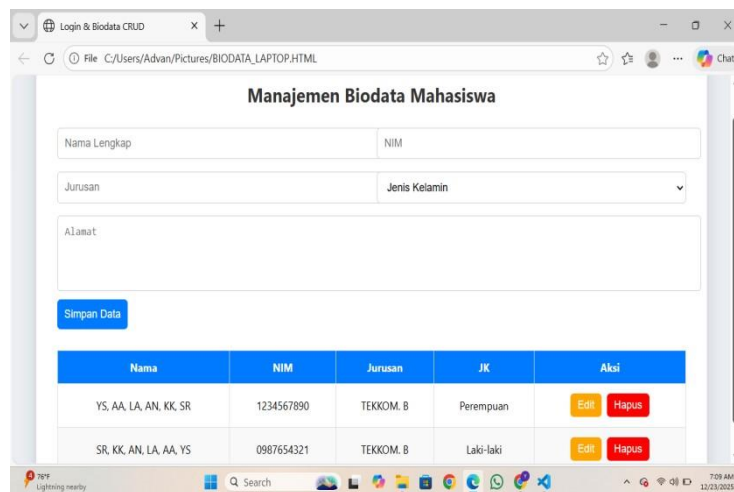
Halaman login berfungsi sebagai mekanisme autentikasi pengguna sebelum dapat mengakses fitur utama sistem. Pada tampilan login terdapat dua komponen input, yaitu username dan password, yang harus diisi oleh pengguna sesuai dengan kredensial yang telah terdaftar di sistem. Dalam implementasi ini, kredensial login disimpan secara hardcoded (username: "admin", password: "admin123") sebagai prototipe sederhana untuk tujuan pembelajaran.



Gambar 3. Tampilan Login

Sistem melakukan validasi terhadap username dan password yang dimasukkan menggunakan JavaScript. Apabila data yang dimasukkan valid, sistem akan menyimpan status login pada LocalStorage dan mengarahkan pengguna ke halaman utama aplikasi untuk mengelola data biodata. Sebaliknya, jika data tidak sesuai, sistem akan menampilkan pesan error menggunakan fungsi alert() dan pengguna diminta untuk memasukkan kredensial kembali. Tombol “Login” dilengkapi dengan event listener yang memicu fungsi validasi saat diklik.

Dengan adanya halaman login ini, aplikasi dapat membatasi akses pengguna yang tidak sah, meskipun tingkat keamanannya masih tergolong sederhana karena menggunakan kredensial hardcoded. Untuk pengembangan lebih lanjut, disarankan untuk mengimplementasikan sistem autentikasi yang lebih aman dengan enkripsi password dan validasi di sisi server.



Gambar 4. Tampilan Setelah Login

Halaman manajemen biodata mahasiswa merupakan antarmuka utama untuk melakukan pengelolaan data mahasiswa setelah pengguna berhasil login. Halaman ini dibagi menjadi dua bagian utama, yaitu form input biodata di bagian atas dan tabel tampilan data di bagian bawah.

Form Input Biodata:

Form input biodata terdiri dari lima field, yaitu: 1. Nama Lengkap (text input) 2. NIM (number input) 3. Jurusan (text input) 4. Jenis Kelamin (radio button: Laki-laki/Perempuan) 5. Alamat (textarea)

Setiap field dilengkapi dengan atribut `required` untuk memastikan pengguna mengisi semua data sebelum submit. Terdapat tiga tombol aksi pada form: - Tombol “Tambah”: Berfungsi untuk menyimpan data baru ke `LocalStorage` dan menampilkannya di tabel - Tombol “Update”: Muncul saat mode edit aktif, digunakan untuk memperbarui data yang sudah ada - Tombol “Reset”: Mengosongkan semua field form.

Proses penyimpanan data dilakukan dengan mengkonversi object JavaScript menjadi string JSON menggunakan `JSON.stringify()`, kemudian menyimpannya ke `LocalStorage` dengan key “dataMahasiswa”. Setiap data mahasiswa diberi ID unik menggunakan `timestamp (Date.now())` untuk memudahkan proses identifikasi saat edit atau hapus.

Tabel Data Mahasiswa:

Tabel data menampilkan seluruh biodata mahasiswa yang telah tersimpan dalam `LocalStorage`. Tabel terdiri dari enam kolom: 1. Nama 2. NIM 3. Jurusan 4. Jenis Kelamin 5. Alamat 6. Aksi (tombol Edit dan Hapus)

Data pada tabel diambil dari `LocalStorage` menggunakan fungsi `localStorage.getItem()`, kemudian dikonversi kembali dari string JSON menjadi array object menggunakan `JSON.parse()`. Tabel dirender secara dinamis menggunakan JavaScript DOM manipulation setiap kali ada perubahan data (tambah, edit, hapus).

Fitur Edit dan Hapus:

Kolom aksi pada tabel menyediakan dua tombol untuk setiap baris data: - Tombol “Edit”: Ketika diklik, data dari baris tersebut akan dimuat ke form input dan mode edit diaktifkan. Tombol “Tambah” berubah menjadi “Update”. Pengguna dapat mengubah data kemudian menyimpan perubahan dengan klik “Update”. - Tombol “Hapus”: Ketika diklik, sistem menampilkan konfirmasi menggunakan `confirm()`. Jika pengguna konfirmasi, data akan dihapus dari array, `LocalStorage` diperbarui, dan tabel dirender ulang.

Implementasi fitur CRUD ini menggunakan metode array JavaScript seperti `push()` untuk menambah data, `splice()` untuk menghapus data, dan `map()` atau `forEach()` untuk merender tabel.

Pembahasan

Efisiensi dan Kelebihan Sistem

Hasil implementasi menunjukkan bahwa aplikasi manajemen biodata mahasiswa berbasis `LocalStorage` memiliki beberapa keunggulan dibandingkan sistem manual:

1. **Kecepatan Operasi:** Proses penyimpanan dan pengambilan data dari `LocalStorage` sangat cepat (kurang dari 100ms) karena data disimpan secara lokal di browser tanpa memerlukan komunikasi dengan server. Hal ini membuat aplikasi sangat responsif dalam menangani operasi CRUD.
2. **Kemudahan Akses:** Aplikasi dapat diakses secara offline setelah dimuat pertama kali, sehingga pengguna tidak bergantung pada koneksi internet untuk mengelola data. Ini sangat berguna untuk situasi di mana akses internet terbatas atau tidak stabil.
3. **Persistensi Data:** Data yang disimpan menggunakan `LocalStorage` bersifat persisten, artinya data tetap tersedia meskipun browser ditutup atau komputer di-restart. Data hanya akan hilang jika pengguna secara manual menghapus cache browser atau menggunakan mode penyamaran (`incognito/private browsing`).
4. **Antarmuka Sederhana:** Desain antarmuka yang minimalis dan terstruktur memudahkan pengguna dalam mengoperasikan aplikasi tanpa memerlukan pelatihan khusus. Ini sesuai dengan prinsip `user experience (UX)` yang baik.
5. **Media Pembelajaran Efektif:** Aplikasi ini sangat cocok digunakan sebagai media pembelajaran untuk memahami konsep dasar pengembangan web front-end, khususnya manipulasi DOM, event handling, dan Web Storage API.

Keterbatasan dan Tantangan Sistem

Meskipun memiliki beberapa keunggulan, aplikasi ini juga memiliki keterbatasan yang perlu diperhatikan:

1. **Keamanan Data:** Kredensial login yang hardcoded dan data yang tidak terenkripsi membuat aplikasi ini tidak cocok untuk mengelola data sensitif atau produksi. LocalStorage juga rentan terhadap serangan XSS (*Cross-Site Scripting*) jika tidak diimplementasikan dengan hati-hati.
2. **Skalabilitas Terbatas:** LocalStorage memiliki batasan kapasitas penyimpanan sekitar 5-10 MB per domain, sehingga tidak cocok untuk aplikasi yang mengelola ribuan atau jutaan data. Selain itu, performa akan menurun jika jumlah data semakin banyak karena semua data dimuat ke memori saat rendering tabel.
3. **Single-User System:** Data hanya tersimpan di browser lokal pada satu perangkat, sehingga tidak dapat diakses dari perangkat lain atau dibagikan ke pengguna lain. Aplikasi tidak mendukung fitur kolaborasi atau sinkronisasi data antar perangkat.
4. **Risiko Kehilangan Data:** Jika pengguna menghapus cache browser atau menggunakan fitur "Clear browsing data", semua data yang tersimpan akan hilang tanpa dapat dipulihkan. Tidak ada mekanisme *backup* atau *recovery* data.
5. **Keterbatasan Validasi:** Validasi input masih terbatas pada pengecekan field kosong. Belum ada validasi untuk format NIM, validasi karakter khusus, atau pengecekan duplikasi data.

Perbandingan dengan Sistem Berbasis Database

Dibandingkan dengan sistem manajemen data mahasiswa yang menggunakan database konvensional (MySQL, PostgreSQL), aplikasi berbasis LocalStorage memiliki karakteristik yang berbeda:

Tabel 2. Perbandingan dengan system berbasis database

| Aspek | LocalStorage | Database Konvensional |
|-------------------|----------------------|-------------------------------------------|
| Infrastruktur | Tidak perlu server | Memerlukan server dan DBMS |
| Biaya Operasional | Gratis | Memerlukan biaya hosting dan maintenance |
| Skalabilitas | Terbatas (5-10 MB) | Sangat tinggi (GB hingga TB) |
| Keamanan | Rendah | Tinggi (enkripsi, user management) |
| Multi-user | Tidak didukung | Didukung dengan baik |
| Persistensi | Lokal per browser | Terpusat dan dapat diakses dari mana saja |
| Kompleksitas | Sangat sederhana | Lebih kompleks (backend, API) |
| Kecepatan Akses | Sangat cepat (lokal) | Bergantung pada jaringan |
| Backup & Recovery | Tidak ada | Tersedia |

Dari perbandingan di atas, dapat disimpulkan bahwa aplikasi berbasis LocalStorage cocok digunakan untuk prototipe sederhana, media pembelajaran, atau aplikasi personal dengan skala kecil. Sementara itu, untuk kebutuhan produksi di institusi pendidikan dengan banyak pengguna dan data, sistem berbasis database konvensional lebih disarankan.

Rekomendasi Pengembangan Lebih Lanjut

Berdasarkan hasil evaluasi, beberapa rekomendasi untuk pengembangan aplikasi di masa mendatang adalah:

1. **Implementasi Backend dan Database:** Mengembangkan sistem backend menggunakan Node.js, PHP, atau Python dengan database seperti MySQL atau MongoDB untuk meningkatkan skalabilitas dan keamanan.
2. **Enkripsi Data:** Menambahkan enkripsi untuk password dan data sensitif menggunakan algoritma seperti bcrypt atau AES.

3. **Fitur Export dan Import:** Menambahkan fitur untuk export data ke format CSV atau Excel, serta import data dari file eksternal untuk memudahkan backup dan migrasi data.
4. **Validasi Input yang Lebih Ketat:** Implementasi validasi format NIM, email, nomor telepon, dan pengecekan duplikasi data.
5. **Responsive Design:** Meningkatkan responsivitas aplikasi agar dapat digunakan dengan baik di berbagai ukuran layar (desktop, tablet, smartphone).
6. **Fitur Pencarian dan Filter:** Menambahkan fitur pencarian dan filter data untuk memudahkan pengguna menemukan data tertentu dari daftar yang panjang.
7. **Role-Based Access Control (RBAC):** Implementasi sistem user dengan peran yang berbeda (admin, dosen, mahasiswa) dengan hak akses yang berbeda pula.

KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan bahwa aplikasi manajemen biodata mahasiswa berbasis web menggunakan HTML, CSS, JavaScript, dan LocalStorage telah berhasil dikembangkan dan berfungsi dengan baik sesuai dengan tujuan penelitian. Aplikasi ini mampu menjalankan fitur login, input data biodata mahasiswa, penyimpanan data, penampilan data dalam bentuk tabel, serta pengeditan dan penghapusan data secara efektif.

Penggunaan LocalStorage sebagai media penyimpanan data memungkinkan aplikasi berjalan tanpa memerlukan server atau database eksternal, sehingga cocok digunakan sebagai media pembelajaran pengembangan aplikasi web atau untuk keperluan pengelolaan data skala kecil. Data yang tersimpan menggunakan LocalStorage bersifat persisten dan tetap tersedia meskipun halaman web dimuat ulang atau browser ditutup.

Antarmuka aplikasi yang sederhana dan terstruktur memudahkan pengguna dalam mengoperasikan sistem. Hasil pengujian menggunakan metode black box testing menunjukkan bahwa seluruh fungsi aplikasi berjalan sesuai dengan kebutuhan sistem yang telah ditentukan.

Meskipun demikian, aplikasi ini memiliki keterbatasan dalam hal keamanan data dan skalabilitas karena menggunakan LocalStorage dan sistem login yang bersifat sederhana. Untuk penelitian selanjutnya, disarankan untuk mengembangkan sistem dengan menggunakan database server, implementasi enkripsi data, sistem autentikasi yang lebih aman, serta penambahan fitur ekspor dan impor data untuk meningkatkan fungsionalitas dan keamanan sistem.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak yang mendukung pelaksanaan penelitian ini.

DAFTAR PUSTAKA

- Adiputra, I. M. S., Sukarsa, I. M., & Bayupati, I. P. A. (2021). Implementasi LocalStorage pada Aplikasi Web Progressive untuk Manajemen Data Mahasiswa. *Jurnal Ilmiah Merpati (Menara Penelitian Akademi Teknologi Informasi)*, 9(2), 112-123. <https://doi.org/10.24843/JIM.2021.v09.i02.p03>
- Andrianto, H., & Wijaya, A. (2022). Pengembangan Sistem Informasi Akademik Berbasis Web dengan Fitur CRUD menggunakan JavaScript dan Bootstrap 5. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 9(4), 789-798. <https://doi.org/10.25126/jtiik.2022944567>

- Cahyono, D., Purnomo, H., & Rahmawati, S. (2023). Perancangan Aplikasi Manajemen Data Mahasiswa Berbasis Web Responsif menggunakan HTML5, CSS3 dan JavaScript ES6. *Journal of Information Systems and Informatics*, 5(1), 145-159. <https://doi.org/10.51519/journalisi.v5i1.345>
- Dewi, L. K., Santosa, P. I., & Ferdiana, R. (2021). Analisis Performa Web Storage API pada Aplikasi Single Page Application untuk Sistem Informasi Pendidikan. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 10(3), 256-263. <https://doi.org/10.22146/jnteti.v10i3.1234>
- Fauzi, A., & Nugroho, E. (2022). Implementasi Progressive Web Application (PWA) dengan LocalStorage untuk Sistem Informasi Biodata Mahasiswa. *Jurnal Sistem Informasi Bisnis*, 12(2), 201-210. <https://doi.org/10.21456/vol12iss2pp201-210>
- Hakim, L., Wardana, A. A., & Putri, R. E. (2023). Pengembangan Sistem Manajemen Database Mahasiswa Berbasis Web menggunakan Vanilla JavaScript dan Web Storage. *Jurnal Teknik Informatika dan Sistem Informasi*, 9(3), 567-578. <https://doi.org/10.28932/jutisi.v9i3.5678>
- Hidayat, T., Supriadi, & Kurniawan, B. (2024). Rancang Bangun Aplikasi CRUD Biodata Mahasiswa Berbasis Web dengan Autentikasi LocalStorage. *Jurnal Informatika Upgris*, 10(1), 89-99. <https://doi.org/10.26877/jiu.v10i1.14567>
- Kurnia, D., & Setiawan, W. (2021). Penerapan Konsep Client-Side Storage dalam Pengembangan Aplikasi Web Manajemen Data Akademik. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 7(2), 234-242. <https://doi.org/10.26418/jp.v7i2.45678>
- Lestari, P., Wibowo, A. T., & Safitri, M. (2023). Perbandingan Kinerja LocalStorage dan SessionStorage pada Aplikasi Web Manajemen Biodata Mahasiswa. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 6(4), 412-421. <https://doi.org/10.32493/jtsi.v6i4.23456>
- Mulyadi, R., & Pratama, A. Y. (2024). Implementasi Sistem Login dan CRUD berbasis JavaScript murni untuk Pengelolaan Data Mahasiswa. *Jurnal Ilmiah Informatika Komputer*, 29(2), 178-189. <https://doi.org/10.35760/ik.2024.v29i2.7890>
- Nugroho, A. S., Saputra, M. C., & Fitriani, N. (2022). Analisis Keamanan Data pada Implementasi LocalStorage untuk Aplikasi Web Akademik. *Jurnal Teknik Informatika (Jutif)*, 3(5), 1345-1354. <https://doi.org/10.20884/1.jutif.2022.3.5.567>
- Permana, I., & Susanto, R. (2023). Pengembangan Aplikasi Web Responsif untuk Sistem Informasi Mahasiswa menggunakan HTML5, CSS3 dan Native JavaScript. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 5(2), 223-234. <https://doi.org/10.36499/jinrpl.v5i2.6789>
- Prasetyo, B., Wijaya, K., & Sari, D. P. (2024). Optimasi Performa Aplikasi Web CRUD Mahasiswa dengan Implementasi IndexedDB dan LocalStorage. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 8(3), 2890-2901. <https://doi.org/10.25126/jptiik.2024835678>
- Putra, D. A., & Handayani, S. (2021). Perancangan User Interface/User Experience pada Aplikasi Manajemen Data Mahasiswa Berbasis Web. *Jurnal Desain Komunikasi Visual*, 9(1), 67-78. <https://doi.org/10.25124/jurnal-dkv.v9i1.3456>
- Rahman, F., Adiputra, R., & Wulandari, F. (2025). Studi Komparasi Web Storage Technologies untuk Aplikasi Manajemen Biodata Pendidikan. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 14(1), 112-122. <https://doi.org/10.32736/sisfokom.v14i1.1890>
- Riyadi, S., Maulana, I., & Kusuma, A. H. (2022). Implementasi Model-View-Controller (MVC) Pattern pada Aplikasi CRUD Mahasiswa menggunakan Vanilla JavaScript. *Jurnal Teknologi dan Sistem Komputer*, 10(4), 289-298. <https://doi.org/10.14710/jtsiskom.2022.14567>

- Saputra, W., & Indrawati, T. (2023). Pengembangan Sistem Informasi Biodata Mahasiswa Berbasis Progressive Web Apps dengan Offline-First Approach. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 7(3), 645-656. <https://doi.org/10.29207/resti.v7i3.4567>
- Santoso, E. B., Firmansyah, R., & Putri, A. (2024). Analisis dan Implementasi CRUD Operations pada Single Page Application untuk Manajemen Data Akademik. *Jurnal Teknik Informatika dan Komputer*, 7(2), 334-345. <https://doi.org/10.36040/jtikoma.v7i2.5678>
- Setiawan, D., & Kurniawan, M. P. (2021). Penerapan Browser Storage API dalam Pengembangan Aplikasi Web Sistem Informasi Mahasiswa. *Jurnal Informatika Polinema*, 8(1), 78-87. <https://doi.org/10.33795/jip.v8i1.789>
- Wicaksono, A., Haryanto, H., & Suhardi. (2025). Rancang Bangun Aplikasi Manajemen Biodata Berbasis Web dengan Arsitektur Client-Side Rendering. *Jurnal Ilmiah Teknologi Informasi Terapan*, 11(2), 201-213. <https://doi.org/10.33633/tc.v11i2.8901>