

Rancang Bangun Sistem *To-Do List* Berbasis Website dalam Efisiensi Pengelolaan Tugas

Nazwa Praditta^{1*}, Apriade Voutama²

^{1*,2} Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang

*Email: pradittanazwa@gmail.com

Abstract

The advancement of information technology has driven the digitalization of task management. Conventional paper-based task recording methods have limitations in terms of accuracy, accessibility, and data security. This study aims to develop a web-based to-do list system using Agile Development within the System Development Life Cycle (SDLC) framework. The system design utilizes Unified Modeling Language (UML) to model business processes and user interactions. The development process employs React.js for the interface, MongoDB as the database, and testing methods including Black Box Testing and Usability Testing. The results indicate that all key features function according to specifications. Usability testing with 10 respondents yielded an SUS score of 88%, indicating a very high level of usability. Users reported increased efficiency in task recording, updating, and monitoring compared to conventional methods. This system is considered suitable for implementation as a web-based task management tool.

Keywords: To-do list, agile development, react.js, mongodb, usability testing

Abstrak

Perkembangan teknologi informasi mendorong digitalisasi pengelolaan tugas. Metode pencatatan konvensional berbasis kertas memiliki keterbatasan dalam akurasi, aksesibilitas, dan keamanan data. Penelitian ini bertujuan mengembangkan sistem *to-do list* berbasis web menggunakan *Agile Development* dalam kerangka *System Development Life Cycle* (SDLC). Perancangan sistem menggunakan *Unified Modeling Language* (UML) untuk memodelkan proses bisnis dan interaksi pengguna. Pengembangan sistem dilakukan dengan React.js sebagai antarmuka, MongoDB sebagai basis data, serta pengujian menggunakan *Black Box Testing* dan *Usability Testing*. Hasil pengujian menunjukkan seluruh fitur utama berfungsi sesuai spesifikasi. *Usability testing* dengan 10 responden menghasilkan skor SUS sebesar 88%, yang mengindikasikan sistem memiliki tingkat *usability* yang sangat baik. Pengguna merasakan peningkatan efisiensi dalam pencatatan, pembaruan, dan pemantauan tugas dibandingkan metode konvensional. Sistem ini dinilai layak diimplementasikan sebagai alat bantu pengelolaan tugas berbasis website.

Kata kunci: To-do list, agile development, react.js, mongodb, usability testing

1. Pendahuluan

Kemajuan teknologi telah membawa perubahan yang substansial pada berbagai sektor kehidupan, termasuk strategi individu dalam mengelola tugas dan pekerjaan. Namun, masih banyak individu yang mengandalkan metode konvensional, seperti pencatatan tugas di kertas memiliki keterbatasan dalam pembaruan informasi, dan berisiko hilang. Keterbatasan ini dapat menghambat pengelolaan tugas secara optimal, sehingga diperlukan solusi berbasis teknologi untuk meningkatkan efektivitas pencatatan dan pengorganisasian tugas [1].

Pengelolaan tugas yang tidak terstruktur dapat menurunkan efisiensi kerja, menghambat pencapaian target, dan meningkatkan beban kognitif. Metode *to-do list* umum digunakan dalam mengorganisir tugas. Meskipun berbagai sistem digital telah tersedia, keterbatasan dalam aksesibilitas dan integrasi dengan kebutuhan pengguna masih menjadi tantangan. Oleh karena itu, diperlukan sistem *to-do list* berbasis website yang lebih optimal dalam mendukung pengelolaan tugas secara digital.

Transformasi digital sistem *to-do list* berbasis website melalui pendekatan *Agile Development* memberikan solusi yang bersifat adaptif terhadap kebutuhan pengguna sekaligus menjamin aksesibilitas optimal lintas platform, secara efektif mengatasi keterbatasan metode konvensional dan meningkatkan efisiensi pengelolaan tugas secara signifikan.

Penelitian ini bertujuan untuk mengembangkan sistem *to-do list* berbasis website dalam mendukung pencatatan, pengorganisasian, serta pemantauan tugas secara digital dan terstruktur. Dengan memanfaatkan React.js untuk membangun antarmuka yang interaktif [2]. Sementara MongoDB digunakan untuk penyimpanan data secara fleksibel [3]. Implementasi teknologi diharapkan dapat meningkatkan efisiensi pengelolaan tugas yang lebih unggul dibandingkan metode konvensional.

Penelitian serupa dilakukan oleh Ulfah Nuraini dan Kholid Haryono, tahun 2022 terkait Rancang Bangun Sistem Informasi Untuk Meningkatkan Efektivitas dan Efisiensi Manajemen Pengelolaan Kegiatan Ramadhan. Penelitian tersebut mengadopsi pendekatan serupa dalam pengembangan sistem *to-do list* guna mengoptimalkan pengelolaan tugas dan aktivitas kerja. Meskipun lingkungannya berbeda, keduanya menekankan digitalisasi dengan sistem berbasis website yang berkontribusi terhadap peningkatan produktivitas dalam pengelolaan kegiatan [4].

Penelitian ini mengadopsi pendekatan serupa dengan yang dilakukan oleh Praditha Ayu Lestari, tahun 2022 mengenai Aplikasi *Me-List* Berbasis Android Menggunakan *Framework* React Native. Kedua penelitian ini berfokus pada sistem pengelolaan tugas, namun penelitian ini berfokus pada aksesibilitas berbasis website [5].

Penelitian ini berkaitan dengan studi Laila Puspita Sari dan Apriade Voutama, tahun 2024 terkait Rancang Bangun Sistem Informasi Stok Sayuran untuk Pedagang Sayur UMKM Menggunakan *Metode Agile Development*. Keduanya menerapkan *Agile Development*, dan *Unified Modeling Language* (UML) sebagai pemodelan sistem. Namun penelitian rancang bangun sistem *to-do list* berbasis website berfokus pada efisiensi pengelolaan tugas dengan penerapan React.js dan MongoDB [6].

Sistem *to-do list* dikembangkan dengan menerapkan teknologi berbasis website. Website merupakan suatu tempat dalam jangkauan internet, mencakup halaman web yang berperan dalam penyimpanan data serta interaksi informasi [7]. Pada proses perancangannya, diperlukan pendekatan rancang bangun yang sistematis. Rancang bangun merupakan tahapan untuk menjelaskan *output* kajian sistem menjadi representasi program sehingga dijadikan dasar dalam mendefinisikan sistem [8].

Dalam hal perancangan sistem digunakannya alat pemodelan berupa *Unified Modeling Language* (UML) guna mendukung proses desain sistem, sehingga meminimalkan kegagalan pengembangan sistem [9]. UML merupakan standar pemodelan untuk memvisualisasikan arsitektur dan dinamika sistem. Pendekatan ini meliputi pengidentifikasian entitas dan keterkaitan komponennya [10]. Dalam pengembangan antarmuka, React.js digunakan karena mengimplementasikan konsep *Single Page Application* (SPA) yang mengoptimalkan waktu muat halaman website [11]. React.js merupakan *framework* berbasis JavaScript untuk membangun antarmuka pengguna secara efisien [12].

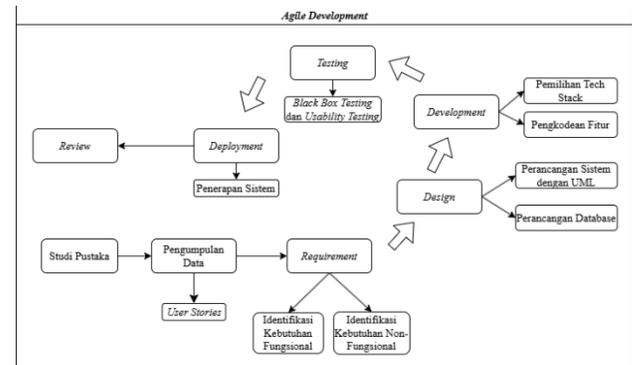
Dari segi manajemen data, MongoDB adalah *database system* NoSQL yang digunakan karena mendukung kapabilitas tinggi dalam pengelolaan data. Dengan menerapkan representasi data dalam format JSON, MongoDB memungkinkan pengelolaan informasi tanpa keterikatan pada struktur tabel tetap [13]. Hal tersebut meningkatkan fleksibilitas pencatatan, pembaruan serta pengelolaan data tugas pada sistem dengan optimal.

Software Development Life Cycle (SDLC) adalah metodologi terstruktur yang umum diterapkan oleh berbagai pengembang *software* untuk menghasilkan sistem yang baik yang mencakup berbagai pendekatan, salah satunya adalah *Agile Development* [14]. Metode *Agile Development* diterapkan pada sistem *to-do list* karena metode ini merupakan pendekatan perancangan sistem dengan waktu singkat yang fleksibel terhadap setiap perubahan yang terjadi [6]. Prinsip dasar *Agile Development* menekankan iterasi dan keterlibatan pengguna yang berkelanjutan untuk menjamin akurasi kebutuhan sistem [15]. Metode ini dipilih karena mendukung proses perancangan sistem yang setiap tahapnya dapat dievaluasi secara berkala sehingga memberikan solusi yang efisien. Efisiensi merupakan tolak ukur pemanfaatan berbagai komponen pada suatu aktivitas [16].

Black Box Testing merupakan metode pengujian yang menilai fungsionalitas sistem berdasarkan kesesuaian antara masukan dan keluaran yang dihasilkan [17]. Sementara itu, *Usability Testing* digunakan untuk mengevaluasi pengalaman pengguna dalam mengelola sistem guna memastikan keterjangkauan efektivitas dan efisiensi penggunaannya [18]. Melalui kombinasi *Black Box Testing* dan *Usability Testing*, sistem diuji baik secara fungsional maupun melalui kemudahan penggunaan dalam kondisi operasional nyata [19].

2. Metode Penelitian

Penelitian ini menerapkan metode *Agile Development* dalam kerangka *System Development Life Cycle* (SDLC), menggunakan langkah sistematis sebagaimana disajikan pada gambar berikut.



Gambar 1. Metode Penelitian *Agile Development*

2.1. Studi Pustaka

Penelitian dimulai melalui studi pustaka guna memperoleh pemahaman yang komprehensif terkait sistem *to-do list* serta penerapan metodologi pengembangannya. Kajian ini mencakup analisis terhadap jurnal ilmiah yang membahas konsep dasar rancang bangun sistem, sistem manajemen tugas seperti *to-do list* serta implementasi *Agile Development*.

2.2. Pengumpulan Data

Tahap pengumpulan data diterapkan melalui penyusunan *User Stories* untuk merinci fitur, aktivitas, dan indikator pencapaian sistem. *User Stories* yang terdokumentasi dengan baik menjadi acuan

dalam pengembangan fitur agar sesuai dengan kebutuhan pengguna pada sistem *to-do list*.

2.3. Requirement

Pada tahap ini dilakukannya identifikasi kebutuhan fungsional yang berkaitan dengan fitur pengelolaan tugas serta kebutuhan non-fungsional yang mencakup aspek kinerja, keamanan, dan aksesibilitas. Menggunakan *User Stories* yang telah disusun, setiap kriteria dianalisis untuk memastikan kesesuaian dengan tujuan sistem.

2.4. Design

Tahap desain diawali dengan perancangan sistem menggunakan alat pemodelan UML untuk merepresentasikan alur kerja sistem *to-do list*. Pemodelan ini mencakup *use case diagram*, *activity diagram*, serta *class diagram*. Setelah pemodelan sistem selesai, selanjutnya perancangan *mockup* serta model *database* sebagai representasi awal tampilan dan struktur penyimpanan data.

2.5. Development

Pada tahap pengembangan, sistem mulai direalisasikan berdasarkan desain yang telah dirancang menggunakan JavaScript sebagai Bahasa pemrograman, dengan *framework* React.js serta MongoDB sebagai basis data. Implementasi mencakup penulisan kode program serta integrasi berbagai komponen sistem.

2.6. Testing

Pada tahap ini dilakukannya pengujian sistem dengan menggunakan metode *Black Box Testing* dan *Usability Testing*. Pengujian *black box* dilakukan dengan menguji berbagai skenario masukan dan menganalisis keluaran tanpa memeriksa kode sumber secara langsung. Apabila ditemukan ketidaksesuaian dalam sistem, maka dilakukan perbaikan untuk memastikan sistem dapat beroperasi secara optimal sebelum diterapkan secara menyeluruh. Sedangkan pengujian *usability* melibatkan 10 responden mahasiswa yang menyelesaikan skenario penggunaan sistem, kemudian menilai pengalaman mereka melalui kuesioner SUS.

2.7. Deployment

Pada tahap *deployment*, sistem yang telah dikembangkan diterapkan secara penuh dan siap digunakan. Proses ini mencakup instalasi, konfigurasi, serta memastikan bahwa sistem *to-do list* dapat beroperasi dengan optimal sesuai kebutuhan pengguna.

2.8. Review

Tahap akhir dilakukannya *review* sebagai evaluasi untuk menilai respon pengguna sistem *to-do list*.

3. Hasil Penelitian

3.1. User Stories

Tabel berikut merangkum *User Stories* beserta kriteria penerimaannya untuk memastikan implementasi sistem sesuai dengan kebutuhan pengguna secara optimal.

Tabel 1. *User Stories* Sistem *To-Do List*

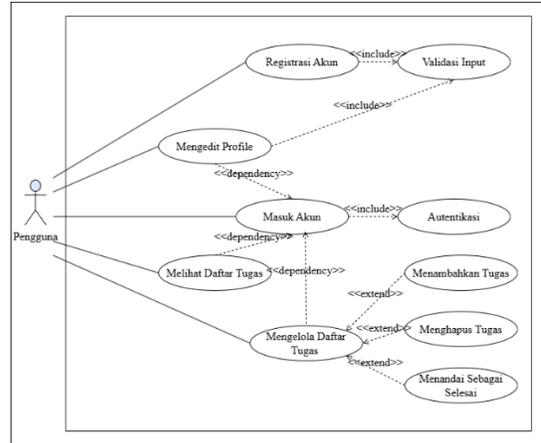
No.	Fitur	User Story	Kriteria Penerimaan
1	Registrasi	Sebagai pengguna baru, saya ingin dapat mendaftar dan membuat akun agar dapat mengelola daftar tugas. Sebagai pengguna, saya ingin masuk pada sistem menggunakan akun yang telah saya daftarkan agar dapat mengakses daftar tugas saya.	Pengguna dapat mengisi formulir registrasi dengan nama, email, dan password yang valid. Setelah berhasil, sistem mengarahkan pengguna ke halaman login.
2	Login	Sebagai pengguna, saya ingin dapat menambahkan tugas baru agar dapat mengelola pekerjaan saya dengan	Pengguna dapat login menggunakan email dan password yang valid. Jika email atau password salah, sistem menampilkan pesan kesalahan.
3	Tambah Tugas	Sebagai pengguna, saya ingin dapat menambahkan tugas baru agar dapat mengelola pekerjaan saya dengan	Pengguna dapat menambahkan tugas yang berhasil ditambahkan akan muncul dalam daftar tugas.

		baik.	
4	Hapus Tugas	Sebagai pengguna saya ingin dapat menghapus tugas yang tidak diperlukan dan tugas yang sudah selesai agar rapi dan terorganisir. Sebagai pengguna saya ingin dapat menandai tugas sebagai selesai agar saya dapat melihat progres pekerjaan saya.	Pengguna dapat memilih tugas yang ingin dihapus. Setelah dihapus tugas tidak muncul lagi dalam daftar tugas. Pengguna dapat mengubah status tugas menjadi "selesai". Tugas yang telah diselesaikan akan dipindahkan ke daftar tugas yang telah selesai.
5	Tandai Tugas Selesai	Sebagai pengguna saya ingin dapat memperbaiki informasi akun seperti nama, email, password, dan foto profil agar data saya tetap akurat dan relevan.	Pengguna dapat mengakses halaman edit profil. Sistem dapat memungkinkan perubahan nama, email, password, dan foto profil. Setelah perubahan disimpan sistem memperbaiki informasi akun pengguna.
6	Edit Profil	Sebagai pengguna saya dapat keluar dari akun saya supaya data saya tetap aman.	Pengguna dapat keluar dari akun dengan menekan tombol logout. Setelah logout, pengguna akan diarahkan kembali ke halaman login.
7	Logout		

3.2. Design

Penelitian ini mengimplementasikan beberapa jenis diagram UML, diantaranya *use case diagram*, *activity diagram*, serta *class diagram*.

3.2.1. Use Case Diagram

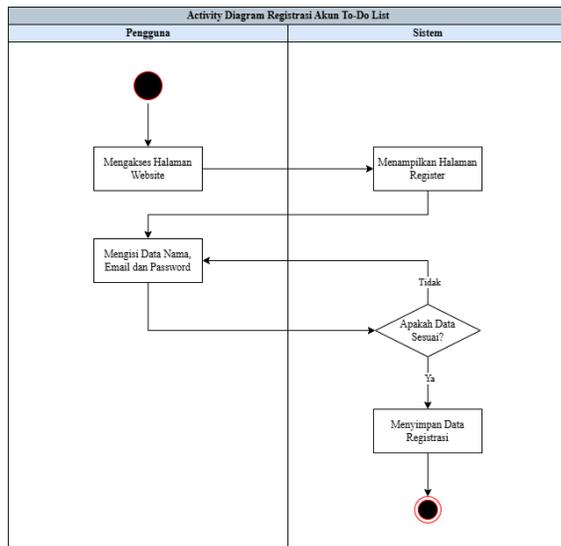


Gambar 2. Use Case Diagram Sistem To-Do List

Pada *use case diagram* sistem *to-do list* merepresentasikan relasi antara pengguna dengan sistem *to-do list*. Pengguna melakukan registrasi akun, mengedit profil, masuk akun, mengelola daftar tugas, dan melihat daftar tugas. Registrasi akun dan pengeditan profil memerlukan proses validasi input, sementara proses masuk akun membutuhkan autentikasi untuk memastikan identitas pengguna. Selain itu, akses ke fitur pengeditan profil, melihat daftar tugas, dan mengelola daftar tugas hanya dapat dilakukan setelah pengguna berhasil masuk ke dalam sistem. Fitur pengelolaan daftar tugas mencakup tiga fungsi utama, yaitu menambahkan tugas, menghapus tugas, dan menandai tugas sebagai selesai.

3.2.2. Activity Diagram

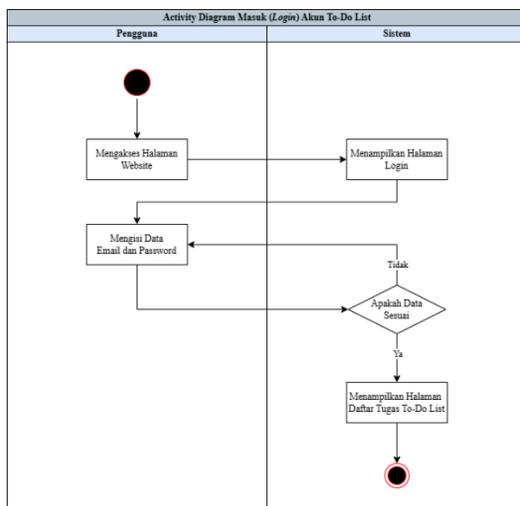
a. Activity Diagram Register Pengguna



Gambar 3. Activity Diagram Registrasi Akun

Pada *activity diagram* sistem *to-do list* menggambarkan tahapan pendaftaran pengguna baru pada sistem *to-do list*. Proses dimulai ketika pengguna mengakses halaman registrasi dan ditampilkan oleh sistem. Pengguna menginput data berupa nama, email, dan password, yang selanjutnya divalidasi oleh sistem. Jika data yang dimasukkan memenuhi ketentuan, sistem akan menyimpan data registrasi pengguna. Namun, jika data tidak sesuai, sistem akan meminta pengguna untuk mengoreksi input yang tidak valid sebelum proses dapat dilanjutkan.

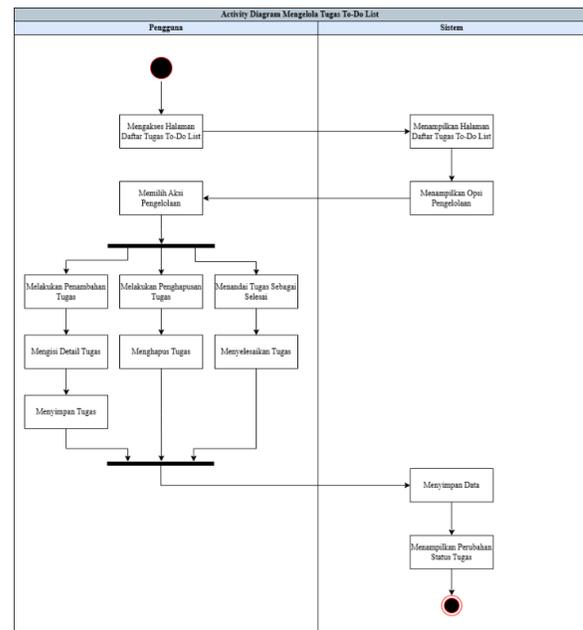
b. Activity Diagram Masuk Akun



Gambar 4. Activity Diagram Masuk Akun

Proses masuk akun diawali dengan pengguna mengakses halaman login, selanjutnya sistem menampilkan *form* login. Selanjutnya, pengguna mengisi email beserta password, dan diverifikasi oleh sistem. Apabila data sesuai dengan yang terdapat di *database*, sistem menampilkan halaman daftar tugas. Sebaliknya, jika autentikasi gagal, sistem akan meminta pengguna untuk menginput ulang data yang benar.

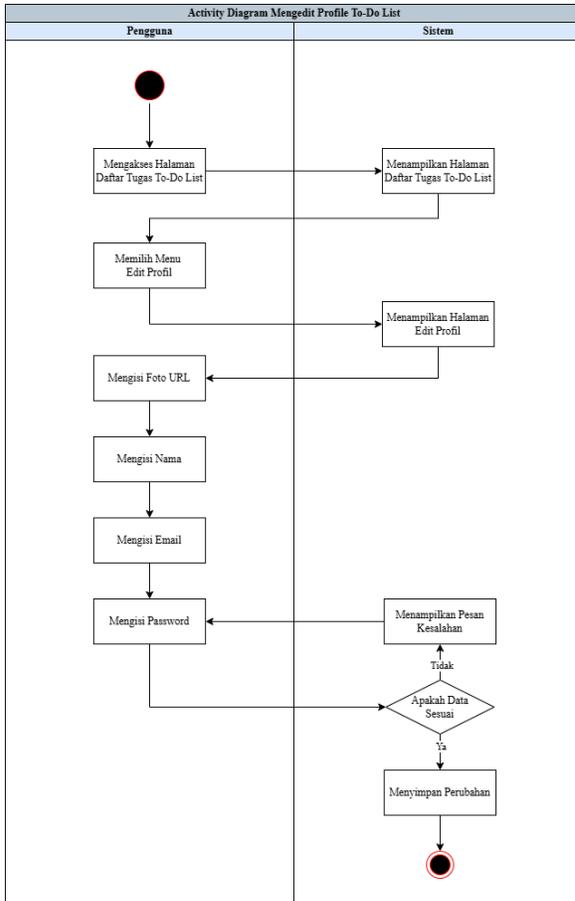
c. Activity Diagram Mengelola Tugas



Gambar 5. Activity Diagram Mengelola Tugas

Pada proses mengelola tugas diawali dengan pengguna mengakses halaman daftar tugas, kemudian ditampilkan oleh sistem beberapa opsi pengelolaan tugas. Pengguna dapat memilih untuk menambahkan tugas baru dengan mengisi detail tugas, menghapus tugas, dan menandai tugas sebagai selesai serta menyimpannya. Setelah salah satu aksi dilakukan, sistem akan menyimpan dan memperbarui data dalam basis data serta menampilkan perubahan status tugas kepada pengguna.

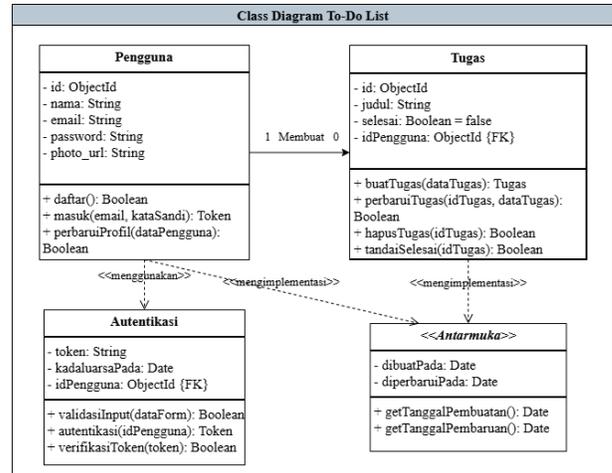
d. *Activity Diagram* Mengedit Profil



Gambar 6. *Activity Diagram* Mengedit Profil

Proses mengedit profil dimulai dengan pengguna mengakses halaman daftar tugas, kemudian terdapat *button* edit profil. Sistem menampilkan halaman edit profil, di mana pengguna dapat memperbarui data terkait URL foto profil, nama, email, dan password. Setelah data diperbarui, sistem memvalidasi guna memastikan kesesuaian format. Jika validasinya berhasil, perubahan akan disimpan dalam basis data. Namun, jika ditemukan kesalahan, sistem menampilkan pesan kesalahan serta pengguna diminta untuk melakukan koreksi sebelum perubahan diproses.

3.2.3. *Class Diagram*



Gambar 7. *Class Diagram* Sistem To-Do List

Class Diagram sistem *to-do list* menggambarkan struktur sistem dengan empat kelas utama yang saling berelasi. Kelas pengguna menyimpan informasi pengguna dan menyediakan metode untuk pendaftaran, autentikasi, serta pembaruan profil. Kelas ini berelasi satu-ke-banyak dengan kelas tugas, yang menyimpan data tugas dan menyediakan metode manipulasi data. Kelas autentikasi mengimplementasikan kelas pengguna untuk mengelola token dan validasi. Sementara itu, kelas antarmuka berfungsi sebagai *interface* bagi kelas tugas untuk mencatat waktu pembuatan dan pembaruan data. Diagram ini menunjukkan arsitektur sistem yang terstruktur dengan pemisahan entitas dan fungsionalitas yang jelas.

3.3. Implementasi Sistem

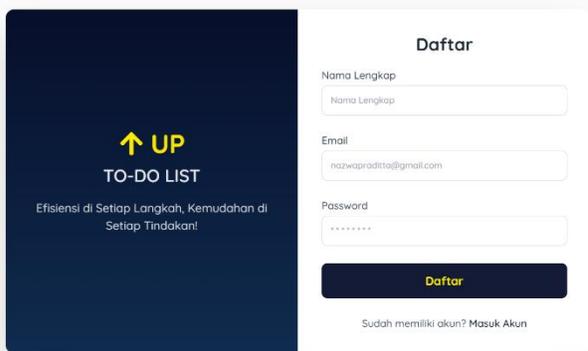
3.3.1. *Tech Stack*

Sistem ini dikembangkan menggunakan JavaScript dengan React.js untuk antarmuka yang dinamis, didukung oleh Tailwind CSS guna memastikan desain yang responsif. Pada sisi *backend*, Node.js dan Express.js digunakan untuk pengelolaan server, sementara MongoDB berperan sebagai basis data No SQL yang fleksibel.

3.3.2. Pengkodean Fitur

Sistem *to-do list* berbasis website ini mencakup beberapa fitur, yaitu fitur autentikasi, kelola tugas, serta kelola profil. Fitur-fitur tersebut diimplementasikan dalam beberapa halaman utama, yaitu halaman register pengguna, halaman login pengguna, halaman kelola tugas dan halaman edit profil.

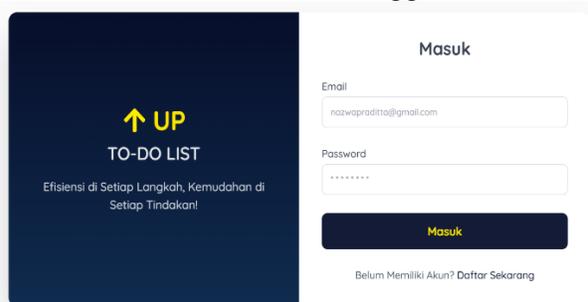
a. Halaman Register Pengguna



Gambar 8. Tampilan Register Pengguna

Halaman register menerapkan fitur autentikasi untuk memastikan keamanan dalam proses pendaftaran akun baru. Pengguna diwajibkan mengisi formulir registrasi yang mencakup nama lengkap, email, dan password. Setelah proses validasi berhasil, akun pengguna akan diaktifkan sehingga dapat mengakses sistem *to-do list*.

b. Halaman Masuk Akun Pengguna

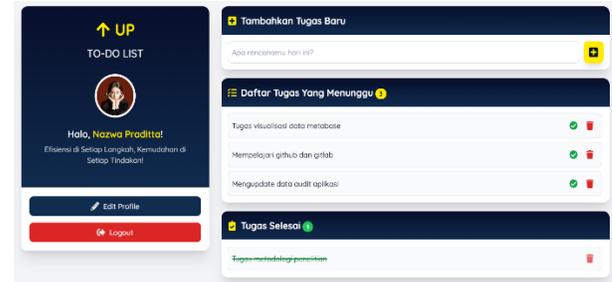


Gambar 9. Tampilan Masuk Akun Pengguna

Halaman masuk akun atau *login* berfungsi sebagai mekanisme autentikasi dalam memastikan bahwa hanya pengguna yang terdaftar saja, yang dapat mengakses sistem *to-do list*. Pengguna wajib memasukkan email beserta password yang sesuai dengan data akun di *database* untuk masuk. Jika validasi

tidak sesuai, maka pengguna tidak dapat masuk ke sistem *to-do list*.

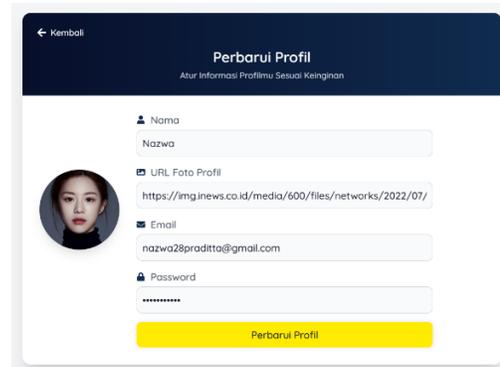
c. Halaman Mengelola Tugas



Gambar 10. Tampilan Mengelola Tugas

Halaman mengelola tugas mengimplementasikan fitur CRUD dengan tambah, lihat daftar tugas, *update*, dan *delete* tugas. Pengguna dapat menambahkan tugas, menghapus tugas jika sudah selesai atau tidak diperlukan, serta mengupdate tugas dengan menandai tugas sebagai selesai. Pada halaman ini juga menampilkan menu untuk mengedit profil pengguna.

d. Halaman Mengedit Profil



Gambar 11. Tampilan Mengedit Profil

Halaman ini mendukung fitur *update* untuk kelola profil, dengan memperbarui data pribadi pengguna. Pada halaman mengedit profil pengguna dapat mengubah profil termasuk nama, URL foto profil, email, dan password. Setiap perubahan dikonfirmasi dengan menekan *button* "Perbarui Profil" guna memastikan bahwa data yang diperbarui tersimpan di *database*. Jika validasi tidak sesuai maka profil pengguna tidak dapat diperbarui di *database*.

3.4. Testing

3.4.1. Black Box Testing

Tahap berikutnya dalam proses pengembangan sistem memasuki tahap pengujian untuk mengevaluasi kesesuaian fungsionalitas terhadap spesifikasi yang telah ditetapkan. Pengujian dilakukan dengan menerapkan metode *black box testing*.

Tabel 2. Hasil *Black Box Testing*

No	Fitur yang Diuji	Masukan	Keluaran	Status
1	Registrasi	Pengguna mendaftar	Akun berhasil dibuat	Berhasil
2	Login	Pengguna masuk dengan kredensial benar	Berhasil masuk	Berhasil
3	Tambah Tugas	Pengguna menambahkan tugas baru	Tugas muncul di daftar	Berhasil
4	Hapus Tugas	Pengguna menghapus tugas	Tugas terhapus	Berhasil
5	Tandai Tugas Selesai	Pengguna menyelesaikan tugas	Tugas berpindah ke daftar tugas selesai	Berhasil
6	Edit Profil	Pengguna mengubah informasi akun	Profil diperbarui	Berhasil
7	Logout	Pengguna keluar dari akun	Kembali ke halaman login.	Berhasil

Berdasarkan hasil pengujian *black box*, seluruh fitur dalam sistem *to-do list* berbasis website ini berfungsi sesuai spesifikasi yang ditetapkan. Temuan ini sejalan dengan penelitian Ulfah Nuraini dan Kholid Haryono [4]. Menunjukkan bahwa sistem berbasis web dapat meningkatkan efektivitas dan efisiensi pengelolaan tugas. Dengan demikian, sistem telah memenuhi standar fungsionalitas dan siap digunakan.

3.4.2. Usability Testing

Setelah sistem *to-do list* berbasis website dikembangkan, dilakukan *usability testing*. Pengujian ini memastikan fitur utama seperti

registrasi, login, pengelolaan tugas, dan edit profil, dapat diakses dengan mudah. Melibatkan 10 mahasiswa sebagai responden, *usability testing* dengan *System Usability Scale* (SUS) mengevaluasi keintuitifan dan efektivitas sistem. Responden berinteraksi langsung, menyelesaikan tugas, dan menilai pengalaman mereka melalui kuesioner SUS, sebagai dasar analisis efisiensi dibandingkan metode konvensional.

Tabel 3. Kuesioner SUS

No	Pernyataan
1	Saya menilai sistem <i>to-do list</i> mudah digunakan.
2	Saya menilai fitur-fitur yang terdapat pada sistem ini memenuhi kebutuhan saya dalam mengelola tugas.
3	Saya merasa dapat menggunakan sistem ini secara langsung tanpa perlu belajar terlebih dahulu.
4	Sistem ini membantu saya dalam menyelesaikan tugas lebih cepat dibandingkan metode konvensional (misalnya mencatat di kertas atau <i>sticky notes</i>).
5	Saya merasa sistem ini memudahkan saya dalam menambahkan, menghapus, serta memperbarui tugas dibandingkan metode konvensional.
6	Saya merasa sistem ini meningkatkan produktivitas saya dalam mengelola tugas sehari-hari.
7	Saya merasa sistem ini mengurangi risiko kehilangan atau lupa tugas dibandingkan dengan pencatatan yang manual.
8	Saya merasa navigasi dalam sistem ini intuitif dan tidak membingungkan.
9	Saya lebih memilih menggunakan sistem ini dibandingkan metode konvensional dalam mengelola tugas saya.
10	Saya merasa sistem ini cukup efisien dan tidak membutuhkan waktu lama dalam penggunaannya.

Tabel 4. Hasil Penilaian *Usability Testing* dengan SUS

No	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10
R 1	4	4	4	4	4	3	4	3	3	3
R 2	4	3	1	3	4	3	3	3	4	3
R 3	4	3	4	3	4	1	4	3	4	4
R 4	4	4	4	4	3	4	4	4	4	4
R 5	4	2	2	4	4	4	3	3	3	4
R 6	4	4	4	4	3	2	4	3	4	4
R 7	4	4	3	3	3	3	4	4	4	4
R 8	4	4	2	4	4	4	4	4	4	4
R	4	4	3	4	3	2	4	4	4	4

9										
R	3	4	3	4	3	4	3	4	3	2
T	Skor Rata-rata								88,0	
Dapat dihitung skor akhir SUS menggunakan persamaan berikut:										

$$NA = \frac{\text{Average}(\text{total skor})}{\text{Jumlah Responden}} \times 100\%$$

$$NA = \frac{88,0}{10} \times 100\%$$

$$NA = 88\%$$

Hasil dari *System Usability Scale* (SUS) yang dilakukan terhadap 10 responden mahasiswa, diperoleh skor rata-rata SUS sebesar 88%. Nilai tersebut menunjukkan bahwa sistem *to-do list* berbasis website memiliki tingkat *usability* yang sangat baik berdasarkan skala interpretasi SUS. Hasil ini sejalan dengan penelitian Praditha Ayu Lestari [5]. Menemukan bahwa sistem manajemen tugas berbasis digital dengan aplikasi meningkatkan aksesibilitas dan kemudahan pengguna.

4. Kesimpulan

Berdasarkan penerapan serta pengujian yang telah dilaksanakan, sistem *to-do list* berbasis website dengan metode *Agile Development* berhasil dikembangkan. Sistem ini dibangun dengan JavaScript dan React.js untuk menciptakan antarmuka yang interaktif dan responsif, serta MongoDB sebagai basis data yang mendukung fleksibilitas pengelolaan data tugas.

Hasil *black box testing* menunjukkan bahwa seluruh fitur utama sistem meliputi registrasi, login, pengelolaan tugas, dan edit profil berfungsi sesuai dengan spesifikasi yang ditetapkan dalam *User Stories*. Sistem ini menjadi solusi digital yang mengatasi keterbatasan metode konvensional, seperti risiko kehilangan catatan dan keterbatasan pembaruan informasi.

Selain itu, terbukti dari *usability testing* menggunakan *System Usability Scale* (SUS) telah memperoleh skor sebesar 88%, mengindikasikan sistem *to-do list* ini memiliki tingkatan *usability* yang “sangat baik”. Skor

tersebut mencerminkan bahwa sistem *to-do list* mudah digunakan, efisien, dan mendukung produktivitas pengguna dalam mengelola tugas.

Dari aspek efisiensi, sistem ini menghemat waktu pencatatan dan pengelolaan tugas dibandingkan metode konvensional. Hasil *usability testing* menunjukkan bahwa sistem mempercepat proses pencatatan, pembaruan, dan pemantauan tugas serta meminimalkan risiko kehilangan catatan. Berdasarkan kombinasi pengujian, sistem ini layak diimplementasikan sebagai alat bantu pengelolaan tugas berbasis website.

5. Saran

Penelitian selanjutnya disarankan untuk memperluas cakupan responden guna memperoleh data yang lebih representatif. Evaluasi sistem dalam skala yang lebih besar juga perlu dilakukan untuk menguji efektivitas sistem secara lebih komprehensif. Sehingga diharapkan dapat memberikan wawasan lebih mendalam terkait kinerja sistem dan potensinya untuk pengembangan lebih lanjut.

6. Daftar Pustaka

- [1] T. Santhi *et al.*, “Pengembangan Sistem Informasi To Do List Berbasis Website dalam Meninjau Kegiatan Mahasiswa Undiksha,” *J. Teknol. Ilmu Komput.*, vol. 1, no. 1, pp. 30–36, 2022.
- [2] R. D. Pradipta *et al.*, “Pengembangan Dashboard Admin menggunakan React JS dan Ant Design pada Toko Berkat,” vol. 6, no. 1, pp. 120–128, 2025.
- [3] J. Penerapan, T. Informasi, B. Perhiasan, B. Flask, and D. A. N. Mongodb, “IT-EXPLORE,” vol. 03, pp. 162–176, 2024.
- [4] H. Falah, U. Nuraini, and K. Haryono, “Rancang Bangun Sistem Informasi Untuk Meningkatkan Efektivitas dan Efisiensi Manajemen Pengelolaan Kegiatan Ramadhan,” *Automata*, vol. 3, no. 2, pp. 1–9, 2022, [Online]. Available: <https://journal.uui.ac.id/AUTOMATA/article/view/24115>
- [5] P. A. Lestari and A. H. Masitoh, “Aplikasi

- Me-List Berbasis Android Menggunakan Framework React Native,” *J. Esensi Infokom J. Esensi Sist. Inf. dan Sist. Komput.*, vol. 6, no. 2, pp. 26–30, 2022, doi: 10.55886/infokom.v6i2.501.
- [6] L. P. Sari *et al.*, “Sayur Umkm Menggunakan Metode Agile Development,” vol. 8, no. 3, pp. 3151–3159, 2024.
- [7] R. Y. Ekadianti, A. Voutama, and A. A. Ridha, “Rancang Bangun Sistem Informasi Pendaftaran Pasien Berbasis Website di Rumah Sakit Permata,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 8, no. 3, p. 249, 2024, doi: 10.30998/string.v8i3.17552.
- [8] M. Fernandes and A. Voutama, “Rancang Bangun Aplikasi Massive Online Open Course Teknologi Digital Berbasis Website Menggunakan Metode Scrum,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3285–3292, 2024, doi: 10.36040/jati.v8i3.9484.
- [9] I. Santosa, L. Siantar, and A. Voutama, “Berbasis Website Pemesanan Tiket Konser Dengan Menerapkan Pemodelan Unified Modeling Language,” vol. 8, no. 6, pp. 12254–12260, 2024.
- [10] A. Voutama, “Sistem Antrian Cuci Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML,” *Komputika J. Sist. Komput.*, vol. 11, no. 1, pp. 102–111, 2022, doi: 10.34010/komputika.v11i1.4677.
- [11] L. O. M. I. Setiawan, Gunawan, and A. Tenriawaru, “Rancang Bangun Sistem Informasi Monitoring Bimbingan Tugas Akhir Berbasis Web Menggunakan Framework Reactjs,” *AnoaTIK J. Teknol. Inf. dan Komput.*, vol. 2, no. 1, pp. 39–50, 2024, doi: 10.33772/anoatik.v2i1.25.
- [12] M. B. Pramadipta, “Rancang Bangun Frontend Website Untuk Pemungutan Suara Dengan Menggunakan React.Js,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4173.
- [13] A. I. Musyaffa, Mulki Indana Zulfa, and Muhammad Syaiful Alim, “Rancang Bangun Purecompute Platform E-Commerce Untuk Belanja Laptop Berbasis Website,” *J. SINTA Sist. Inf. dan Teknol. Komputasi*, vol. 1, no. 1, pp. 21–29, 2024, doi: 10.61124/sinta.v1i1.9.
- [14] I. Yudiantmoko and A. Voutama, “Perancangan Aplikasi Massive Open Online Courses (MOOCs) Berbasis Android Untuk Meningkatkan Keterampilan Digital,” vol. 9503, pp. 123–134, 2024.
- [15] J. N. B. Mendoza, D. G. Buhat-Mendoza, J. O. Ampaguey, and M. G. Gravides, “Leveraging Software Engineering Practices to Enhance Nursing Informatics Systems Development: A Systematic Literature Review,” *2024 Conf. Inf. Commun. Technol. Soc. ICTAS 2024 - Proc.*, pp. 214–220, 2024, doi: 10.1109/ICTAS59620.2024.10507143.
- [16] Tuzahra, Wahyuni, and Maison, “Pengaruh Akuntabilitas, Transparansi Dan Partisipasi Stakeholders Terhadap Efisiensi Pengelolaan Dana Pendidikan (Studi Kasus di Sekolah Dasar Negeri Se-Kecamatan IV Nagari, Kabupaten Sijunjung,” *J. Innov. Res. Knowl.*, vol. 2, no. 5, pp. 2313–2330, 2022.
- [17] F. Tella, R. Soekarta, and ..., “Rancang Bangun Sistem Informasi Manajemen Arsip Stnk Berbasis Web Pada Kantor Samsat Aimas Kabupaten Sorong,” *Insect (Informatics ...*, vol. 09, no. 01, pp. 29–39, 2023.
- [18] S. T. Pernanda *et al.*, “Rancang Bangun Sistem Informasi Pengelolaan Project-Based Learning Dengan Metode Prototyping,” 2020.
- [19] A. Rahmat and N. A. M. Hanifah, “Usability Testing in Kanban Agile Process for Club Management System,” *6th Int. Conf. Interact. Digit. Media, ICIDM 2020*, no. Icidm, 2020, doi: 10.1109/ICIDM51048.2020.9339668.