

**PENGEMBANGAN SISTEM PEMBERIAN TUGAS MAHASISWA DENGAN PENDEKATAN SDLC****Maliki <sup>1\*</sup>, Anggi Perwitasari <sup>2</sup>, Herry Sujaini <sup>3</sup>**<sup>1,2,3</sup> Universitas Tanjungpura; Jl. Daya Nasional Komp.  
Universitas Tanjungpura Pontianak.; +62 82159899734**Abstract (English)**

*Technological advancements have simplified the process of assigning tasks to students, particularly by improving accessibility between lecturers and students. However, the current system allows lecturers to assign tasks independently without coordination with other course lecturers, potentially leading to a buildup of assignments at the same time. This situation makes it difficult for students to complete assignments properly and on time. Therefore, a task management system is needed that enables lecturers to view the scheduling of assignments from other courses as a consideration when assigning new tasks. The system development follows the SDLC methodology using the waterfall model, which includes analysis, design, implementation, and evaluation phases. The system assists lecturers in managing assignments and assessments, and helps students receive and submit their assignments more efficiently. It was developed using the PHP programming language with the CodeIgniter framework. To ensure functionality, testing was carried out using black box testing and user acceptance testing (UAT). The results show that all system functions operate according to specifications, and UAT yielded a score of 91.97%, indicating that the system is considered successful and acceptable to users.*

**Article History**

*Submitted: 9 Juli 2025  
Accepted: 12 Juli 2025  
Published: 13 Juli 2025*

**Key Words**

Algorithm a;  
B algorithms;  
Complexity.

**Abstrak (Indonesia)**

Kemajuan teknologi telah mempermudah proses pemberian tugas kepada mahasiswa, khususnya dalam meningkatkan aksesibilitas antara dosen dan mahasiswa. Namun, sistem saat ini memungkinkan dosen memberikan tugas tanpa koordinasi dengan dosen mata kuliah lain, sehingga dapat menyebabkan penumpukan tugas dalam waktu yang bersamaan. Kondisi ini menyulitkan mahasiswa menyelesaikan tugas dengan baik dan tepat waktu. Oleh karena itu, dibutuhkan sistem pemberian tugas yang memungkinkan dosen memantau jadwal tugas dari mata kuliah lain sebagai bahan pertimbangan dalam menjadwalkan tugas baru. Pengembangan sistem ini menggunakan metode SDLC dengan model waterfall, yang mencakup tahap analisis, perancangan, implementasi, dan evaluasi. Sistem ini membantu dosen dalam pengelolaan tugas dan penilaian, serta memudahkan mahasiswa dalam menerima dan mengumpulkan tugas. Sistem dikembangkan menggunakan bahasa pemrograman PHP dengan framework CodeIgniter. Untuk memastikan fungsionalitas, dilakukan pengujian menggunakan metode black box testing dan user acceptance testing (UAT). Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan sesuai spesifikasi, dan UAT menghasilkan nilai sebesar 91,97%, yang berarti sistem dinilai berhasil serta dapat diterima oleh pengguna.

**Sejarah Artikel**

*Submitted: 9 Juli 2025  
Accepted: 12 Juli 2025  
Published: 13 Juli 2025*

**Kata Kunci**

Algorithm a;  
B algorithms;  
Complexity.

**1. PENDAHULUAN**

Perkuliahan adalah kegiatan akademik atau proses belajar mengajar mahasiswa yang dibimbing oleh dosen dalam bentuk kuliah atau ceramah, seminar, diskusi, praktikum, ataupun tugas tertentu pada perguruan tinggi. Dalam satu perguruan tinggi terdapat berbagai fakultas, jurusan, dan program studi. Masing-masing program studi memiliki mata kuliah tersendiri yang harus diselesaikan oleh mahasiswa. Salah satu proses yang harus dilalui mahasiswa dalam menyelesaikan mata kuliah adalah mahasiswa harus mengerjakan tugas-tugas yang diberikan dosen di setiap mata kuliah. Tugas berguna untuk pendalaman materi pembelajaran oleh mahasiswa yang dirancang oleh dosen untuk mencapai standar kompetensi. Pemberian tugas mahasiswa memiliki serangkaian tahapan yang penting yaitu dimulai dari dosen merancang dan membuat tugas. Kemudian tugas tersebut diberikan kepada mahasiswa. Tugas selanjutnya

diterima dan dikerjakan oleh mahasiswa. Mahasiswa mengumpulkan tugas yang telah dikerjakan. Dosen menerima tugas mahasiswa dan melakukan penilaian serta memberikan umpan balik kepada mahasiswa.

Kemajuan teknologi mempengaruhi kegiatan pemberian tugas mahasiswa. Dosen kerap memberikan informasi tugas kepada mahasiswa melalui platform digital seperti Google Classroom, E-Learning, atau aplikasi WhatsApp. Penggunaan platform digital ini bermanfaat untuk memuat informasi tugas yang dapat diakses oleh mahasiswa secara daring. Informasi tugas biasanya tersimpan di dalam sebuah file dokumen dan diunggah ke platform digital yang digunakan oleh dosen pemberi tugas. Salah satu informasi yang penting terkait pemberian tugas yaitu informasi jadwal tugas. Jadwal tugas merujuk pada waktu pemberian tugas dan deadline tugas. Waktu pemberian tugas mengacu pada saat tugas diberikan kepada mahasiswa yang bertanggung jawab untuk menyelesaikannya. Sedangkan deadline adalah batas waktu akhir dimana tugas harus diselesaikan dan dikumpulkan.

Penelitian yang dilakukan oleh Randi Afif dkk. (2025), tentang aplikasi manajemen tugas mahasiswa, menjelaskan aplikasi yang dibangun berfungsi untuk dosen membuat dan mendistribusikan tugas harian kepada mahasiswa berisi langkah kerja dan deadline. Mahasiswa dapat mengumpulkan tugas melalui aplikasi tersebut. Status dan evaluasi akan diperbarui dosen setelah tugas dikumpulkan oleh mahasiswa [1].

Pemberian tugas dilakukan oleh masing-masing dosen pengampu mata kuliah tanpa adanya keterlibatan antara jadwal pemberian tugas mata kuliah satu dengan jadwal pemberian tugas mata kuliah lainnya. Hal tersebut dapat mengakibatkan adanya penumpukan tugas yang diterima mahasiswa pada waktu tertentu. Berdasarkan hasil kuesioner terhadap 40 mahasiswa mengenai penumpukan tugas di waktu tertentu yaitu tugas mata kuliah dari dosen, didapatkan hasil sebesar 82,5% responden atau mahasiswa menunjukkan pernah mengalami penumpukan tugas mata kuliah yang diberikan oleh dosen. Oleh sebab itu dibutuhkan sebuah sistem pemberian tugas mahasiswa, dimana dosen dapat melihat pemberian tugas yang telah dilakukan oleh dosen lain, sebagai bahan pertimbangan dosen tersebut untuk menentukan pemberian tugas baru agar dapat mencegah terjadinya penumpukan tugas terhadap mahasiswa.

Sebuah sistem yang dikembangkan harus mengikuti langkah atau tahapan tertentu. Salah satu metode untuk mengembangkan sistem yaitu metode System Development Life Cycle (SDLC). Metode SDLC terdiri dari tahapan analisa sistem, spesifikasi kebutuhan sistem, perencanaan sistem, pengembangan sistem, pengujian sistem, dan pemeliharaan sistem [2]. Model SDLC yang diterapkan dalam penelitian ini adalah model waterfall, yang merupakan jenis pendekatan yang linear, bergerak dari atas ke bawah, dimulai dari tahap perencanaan hingga tahap pemeliharaan [3].

Berdasarkan uraian yang telah dijabarkan sebelumnya, menjadi dasar penulis untuk melakukan penelitian tentang Pengembangan Sistem Pemberian Tugas Mahasiswa dengan Pendekatan SDLC sebagai platform digital pemberian tugas mahasiswa yang dapat digunakan oleh dosen dan mahasiswa. Sistem nantinya dapat menampilkan jadwal tugas mata kuliah lainnya disaat dosen akan memberikan tugas baru. Selain itu, sistem dapat mengelola data tugas, pengumpulan, dan penilaian terhadap tugas mahasiswa. Sistem akan dibangun dalam bentuk website agar mudah diakses oleh dosen dan mahasiswa menggunakan web browser secara daring.

## 2. TINJAUAN PUSTAKA

### 2.1. Sistem

Sistem pada dasarnya merupakan sekumpulan komponen yang saling terhubung dan bekerja secara terpadu guna mencapai tujuan tertentu. Dalam konteks sistem informasi, komponen tersebut mencakup unsur manusia, perangkat keras, perangkat lunak, data, serta jaringan, yang

seluruhnya berinteraksi untuk mendukung proses pengumpulan, pengolahan, penyimpanan, dan distribusi informasi secara efisien [4].

## 2.2. Tugas

Tugas mahasiswa adalah aktivitas akademik yang diberikan oleh dosen untuk mengembangkan pemahaman, keterampilan, dan sikap mahasiswa. Tugas ini dirancang agar mahasiswa dapat menerapkan teori, meningkatkan kemampuan analisis, serta mempersiapkan diri untuk tantangan profesional secara mandiri maupun kolaboratif [5].

## 2.3. System Development Life Cycle (SDLC)

SDLC (*System Development Life Cycle*) adalah kerangka kerja yang sistematis untuk mengelola seluruh tahapan pengembangan sistem informasi, mulai dari analisis kebutuhan, perancangan, pengembangan, hingga pemeliharaan. Tujuannya agar proses pengembangan berjalan terstruktur, efisien, dan menghasilkan sistem yang sesuai dengan kebutuhan pengguna [6].

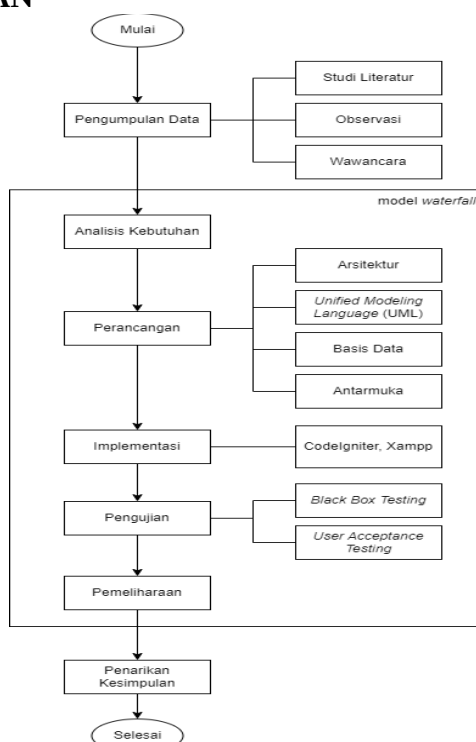
## 2.4. Model Waterfall

Model *Waterfall* adalah metode pengembangan sistem yang mengikuti pendekatan linear dan berurutan, di mana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Tahapan utama meliputi analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Model ini cocok untuk proyek dengan kebutuhan yang jelas dan tidak banyak perubahan selama proses pengembangan [7].

## 2.5. Unified Modelling Language (UML)

UML (*Unified Modeling Language*) adalah bahasa pemodelan standar yang digunakan untuk merancang dan mendokumentasikan sistem perangkat lunak secara visual. UML memudahkan komunikasi antara pengembang dan pemangku kepentingan dengan menyediakan berbagai diagram yang menggambarkan struktur dan perilaku sistem secara terperinci [8].

## 3. METODE PENELITIAN



Gambar 1. Metodelogi Penelitian

### 3.1. Pengumpulan Data

Pengumpulan data menggunakan tiga metode yaitu studi literatur, wawancara, dan observasi.

#### 3.1.1. Wawancara

Semua **Tabel** dan **Gambar** yang anda masukkan dalam dokumen harus jelas terbaca.

#### 3.1.2. Observasi

Jika anda menggunakan *Word*, gunakan persamaan Microsoft Equation Editor atau *MathT*

#### 3.1.3. Studi Literatur

Pengumpulan data dilakukan melalui studi literatur terhadap jurnal-jurnal ilmiah penelitian terkait [1],[9],[10],[11].

### 3.2. Analisis Kebutuhan

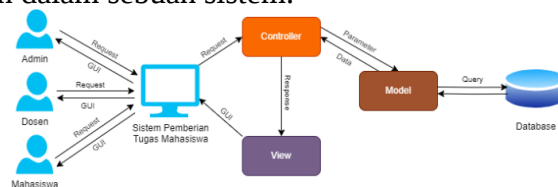
Tahapan analisis kebutuhan merupakan proses awal dalam pengembangan sistem yang bertujuan mengidentifikasi dan mendefinisikan kebutuhan pengguna. Proses ini meliputi pengumpulan informasi (*elicitation*), analisis dan spesifikasi kebutuhan, validasi, serta pengelolaan perubahan kebutuhan sepanjang proyek, guna memastikan sistem yang dibangun sesuai dengan harapan pihak terkait [12].

### 3.3. Perancangan

Perancangan adalah tahap dalam pengembangan sistem informasi di mana solusi teknis dirumuskan berdasarkan kebutuhan yang telah dianalisis. Aktivitasnya mencakup pembuatan desain logis dan fisik, seperti diagram arsitektur sistem, desain basis data, antarmuka pengguna, serta spesifikasi teknis yang digunakan untuk implementasi sistem [13].

#### 3.3.1. Arsitektur

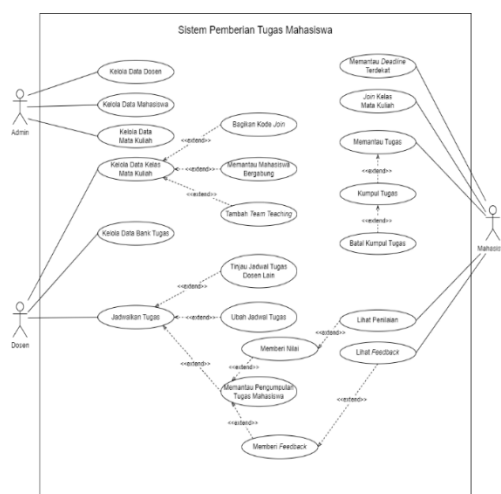
Perancangan arsitektur sistem digunakan untuk menggambarkan komponen utama dan interaksi antar komponen dalam sebuah sistem.



Gambar 2. Arsitektur Sistem

#### 3.3.2. Use Case Diagram

Mendeskripsikan interaksi antara aktor dengan sistem [14] dan menggambarkan fitur-fitur yang tersedia bagi pengguna dalam berinteraksi dengan sistem [15].



Gambar 3. Use Case Diagram

[illegible]

### 3.4. Implementasi

Tahap implementasi yaitu fase di mana rancangan sistem diwujudkan menjadi aplikasi atau solusi yang dapat berfungsi secara nyata. Aktivitas utamanya mencakup penulisan kode, pengujian unit dan integrasi, instalasi sistem pada lingkungan produksi, serta pelatihan pengguna. Tujuan utamanya adalah memastikan bahwa sistem yang dibangun memenuhi spesifikasi desain dan siap digunakan oleh pemangku kepentingan [17]. Pada penelitian ini, sistem dibangun dalam bentuk *website* menggunakan bahasa pemrograman PHP dengan framework CodeIgniter, Javascript, CSS dengan framework Bootstrap, dan MySQL untuk manajemen basis data.

### 3.5. Pengujian

### 3.5.1. Black Box Testing

Metode pengujian perangkat lunak yang menilai fungsionalitas sistem berdasarkan input dan output tanpa melihat struktur kode internal. Pengujian ini fokus pada apakah sistem sesuai dengan kebutuhan pengguna [18]. Pengujian dilakukan dengan menjalankan fitur-fitur yang terdapat pada sistem. Hasil pengujian menunjukkan apakah pengujian dengan masukan yang telah ditentukan menghasilkan output yang sesuai dengan hasil yang diharapkan.

### 3.5.2. User Acceptance Testing

*User Acceptance Testing* (UAT) adalah tahap pengujian akhir dalam pengembangan sistem di mana pengguna akhir (*user*) mencoba sistem dalam skenario nyata untuk memastikan bahwa fungsionalitas, antarmuka, dan performa memenuhi kebutuhan bisnis mereka sebelum sistem digunakan secara resmi. UAT bertujuan untuk memvalidasi sistem secara langsung dari sudut pandang pengguna, memastikan bahwa produk tidak hanya bebas dari kesalahan teknis, tetapi juga sesuai dengan harapan dan kebutuhan *end user* [19].

### 3.6. Pemeliharaan

Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya, seperti perbaikan *bug* pada fungsi atau fitur yang ada pada sistem.

#### 4. HASIL DAN PEMBAHASAN

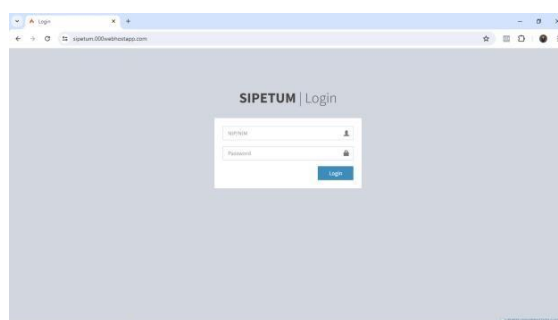
Hasil implementasi yaitu sistem pemberian tugas yang dapat digunakan dosen, mahasiswa, dan admin. Dosen dapat mengelola kelas, mengelola tugas, menjadwalkan tugas, meninjau tugas dosen lain, memeriksa pengumpulan tugas mahasiswa, dan memberikan nilai serta umpan balik. Mahasiswa dapat memonitor tugas dan melakukan pengumpulan tugas. Admin dapat mengelola data pengguna yaitu dosen dan mahasiswa, serta mengelola data mata kuliah.

##### 4.1. Tampilan Antarmuka

Perancangan adalah tahap dalam pengembangan sistem informasi di mana solusi teknis dirumuskan berdasarkan kebutuhan yang telah dianalisis. Aktivitasnya mencakup pembuatan desain logis dan fisik, seperti diagram arsitektur sistem, desain basis data, antarmuka pengguna, serta spesifikasi teknis yang digunakan untuk implementasi sistem [13].

##### 4.1.1. Antarmuka Login

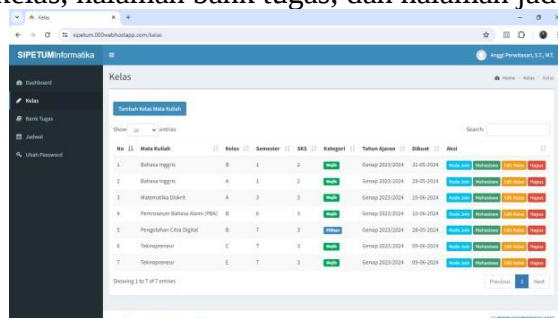
Halaman pertama yang ditampilkan saat sistem diakses. Pengguna harus memasukkan nomor induk dan *password* untuk masuk ke dalam sistem.



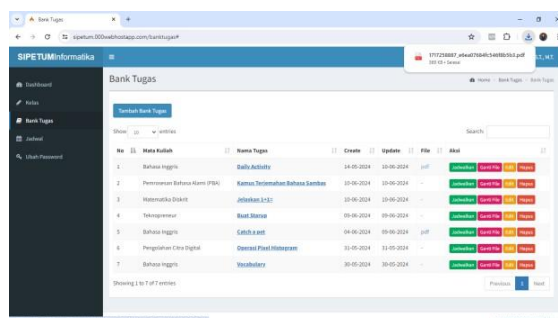
Gambar 5. Antarmuka Halaman Login

##### 4.1.2. Antarmuka Dosen

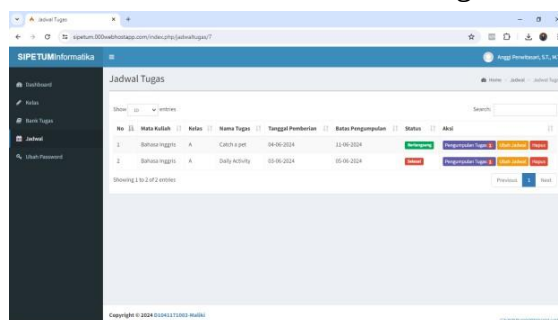
Terdiri dari halaman kelas, halaman bank tugas, dan halaman jadwal.



Gambar 6. Antarmuka Halaman Kelas Dosen



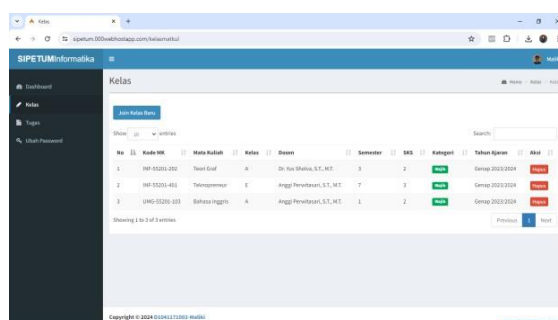
Gambar 7. Antarmuka Halaman Bank Tugas



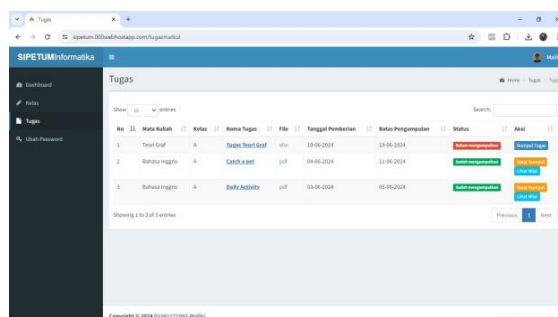
Gambar 8. Antarmuka Halaman Jadwal Tugas

#### 4.1.3. Antarmuka Mahasiswa

Terdiri dari halaman kelas dan halaman tugas.



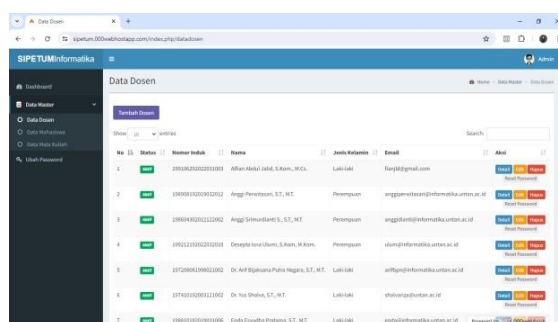
Gambar 9. Antarmuka Halaman Kelas Mahasiswa



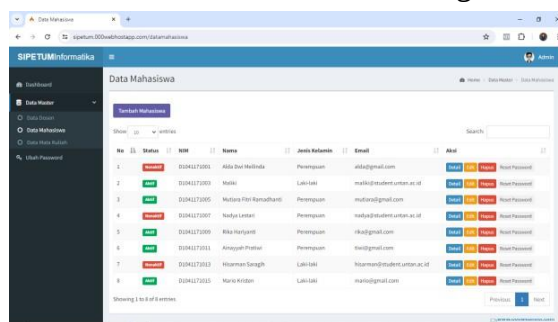
Gambar 10. Antarmuka Halaman Tugas

#### 4.1.4. Antarmuka Admin

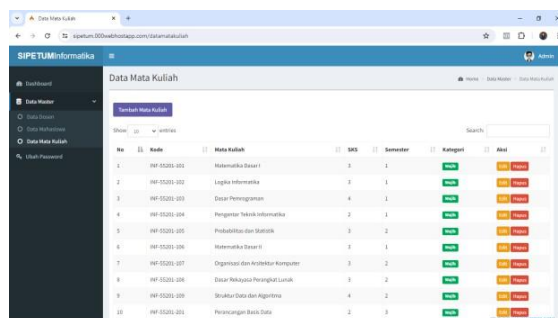
Terdiri dari halaman data dosen, data mahasiswa, dan data mata kuliah.



Gambar 11. Antarmuka Halaman Data Dosen



Gambar 12. Antarmuka Halaman Data Mahasiswa



Gambar 13. Antarmuka Halaman Data Mata Kuliah

#### 4.2. Hasil Black Box Testing

Pengujian *black box* dilakukan dengan menjalankan fitur-fitur yang terdapat pada sistem. Fitur-fitur pada sistem yang telah dibangun diuji dengan melakukan *input* sesuai kasus pengujian yang telah dirancang sebelumnya. Hasil pengujian menunjukkan apakah pengujian dengan masukan yang telah ditentukan menghasilkan output yang sesuai dengan hasil yang diharapkan.

Tabel 1 Hasil Black Box Testing Dosen

| No. | Pengujian                | Masukan                                  | Hasil yang Diharapkan                                       | Hasil  |
|-----|--------------------------|------------------------------------------|-------------------------------------------------------------|--------|
| 1   | Login Pengguna           | Nomor induk dan password benar dan aktif | Pengguna berhasil masuk ke dalam halaman dashboard          | Sukses |
| 2   | Tambah Kelas Mata Kuliah | Semua field diisi dengan benar           | Data kelas mata kuliah tersimpan                            | Sukses |
| 3   | Edit Kelas Mata Kuliah   | Semua field diisi dengan benar           | Perubahan data kelas mata kuliah tersimpan                  | Sukses |
| 4   | Hapus Kelas Mata Kuliah  | Id kelas mata kuliah                     | Data kelas mata kuliah terhapus                             | Sukses |
| 5   | Lihat Kode Join          | Menekan tombol kode <i>join</i>          | Menampilkan kode untuk bergabung ke dalam kelas mata kuliah | Sukses |
| 6   | Tambah Bank Tugas        | Semua field diisi dengan benar           | Data bank tugas tersimpan                                   | Sukses |
| 7   | Edit Bank Tugas          | Semua field diisi dengan benar           | Perubahan data bank tugas tersimpan                         | Sukses |
| 8   | Hapus Bank Tugas         | Id bank tugas                            | Data bank tugas terhapus                                    | Sukses |

| No. | Pengujian         | Masukan                        | Hasil yang Diharapkan                                          | Hasil  |
|-----|-------------------|--------------------------------|----------------------------------------------------------------|--------|
| 9   | Ubah File Tugas   | Format file benar              | Perubahan file tugas tersimpan                                 | Sukses |
| 10  | Jadwalkan Tugas   | Memilih kelas mata kuliah      | Menampilkan jadwal tugas dosen-dosen lain dan form penjadwalan | Sukses |
| 11  | Edit Jadwal Tugas | Semua field diisi dengan benar | Perubahan data jadwal tugas tersimpan                          | Sukses |
| 12  | Hapus Jadwal      | Id jadwal tugas                | Data jadwal tugas terhapus                                     | Sukses |

Tabel 2 Hasil Black Box Testing Mahasiswa

| No. | Pengujian          | Masukan                                   | Hasil yang Diharapkan                                                           | Hasil  |
|-----|--------------------|-------------------------------------------|---------------------------------------------------------------------------------|--------|
| 1   | Join Kelas         | Field diisi kode <i>join</i> dengan benar | Berhasil bergabung dalam kelas mata kuliah dan data kelas mata kuliah tersimpan | Sukses |
| 2   | Hapus Kelas        | Id kelas mata kuliah                      | Data kelas mata kuliah mahasiswa terhapus                                       | Sukses |
| 3   | Kumpulkan Tugas    | Semua field diisi dengan benar            | Data pengumpulan tugas tersimpan                                                | Sukses |
| 4   | Batal Kumpul Tugas | Id kumpul tugas                           | Data pengumpulan tugas terhapus                                                 | Sukses |

Tabel 3 Hasil Black Box Testing Admin

| No. | Pengujian              | Masukan                               | Hasil yang Diharapkan                | Hasil  |
|-----|------------------------|---------------------------------------|--------------------------------------|--------|
| 1   | Tambah Dosen           | Semua <i>field</i> diisi dengan benar | Data dosen tersimpan                 | Sukses |
| 2   | Edit Data Dosen        | Semua <i>field</i> diisi dengan benar | Perubahan data dosen tersimpan       | Sukses |
| 3   | Hapus Data Dosen       | Id dosen                              | Data dosen terhapus                  | Sukses |
| 4   | Tambah Mahasiswa       | Semua <i>field</i> diisi dengan benar | Data mahasiswa tersimpan             | Sukses |
| 5   | Edit Data Mahasiswa    | Semua <i>field</i> diisi dengan benar | Perubahan data mahasiswa tersimpan   | Sukses |
| 6   | Hapus Data Mahasiswa   | Id mahasiswa                          | Data mahasiswa terhapus              | Sukses |
| 7   | Tambah Mata Kuliah     | Semua <i>field</i> diisi dengan benar | Data mahasiswa tersimpan             | Sukses |
| 8   | Edit Data Mata Kuliah  | Semua <i>field</i> diisi dengan benar | Perubahan data mata kuliah tersimpan | Sukses |
| 9   | Hapus Data Mata Kuliah | Id mata kuliah                        | Data mata kuliah terhapus            | Sukses |

### 4.3. Hasil User Acceptance Testing

User Acceptance Testing dilakukan dengan menjelaskan setiap fitur yang ada pada sistem kepada penguji. Penguji mencoba langsung sistem yang telah dibangun. Selanjutnya penguji diberikan pertanyaan-pertanyaan dalam bentuk kuesioner likert summated rating (LSR) untuk memberikan penilaian terhadap tiga aspek yaitu rekayasa perangkat lunak, fungsionalitas, dan komunikasi visual. Hasil pengujian menunjukkan apakah sistem yang dibangun dinilai berhasil atau tidak. Hasil UAT mendapatkan persentase 96% untuk pengguna admin, 89,41% untuk pengguna dosen, dan 90,5% untuk pengguna mahasiswa.

## 5. KESIMPULAN

Setelah dilakukan pengujian dan analisis terhadap hasil penelitian yang dilakukan pada sistem pemberian tugas mahasiswa dengan pendekatan SDLC dapat disimpulkan bahwa penulis berhasil membuat sistem pemberian tugas mahasiswa yang dapat melakukan proses kelola data tugas, data penjadwalan, data pengumpulan tugas, serta penilaian dan *feedback* oleh dosen untuk mahasiswa. Sistem juga menyediakan fitur dimana dosen dapat melihat jadwal tugas dosen lain sebelum melakukan penjadwalan tugas.

Berdasarkan pengujian *black box* yang dilakukan, seluruh fungsi yang ada dalam sistem berhasil dijalankan dengan hasil yang diharapkan. Hasil UAT dari masing-masing bagian pengguna yaitu admin, dosen, dan mahasiswa dengan jumlah responden 15 orang menunjukkan persentase skor 91.97% yang menunjukkan bahwa sistem dinilai berhasil dan dapat diterima oleh pengguna.

### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak yang telah memberi dukungan terhadap penelitian ini.

## DAFTAR PUSTAKA

- [1] R. Afif, N. Hadian, R. Mau'ida, M. Mustaqim, A. Safitri, F. F. A., dan Tholabudin, "Rancang bangun aplikasi manajemen tugas menggunakan Laravel," *Jurnal Ilmiah Teknologi Informasi dan Komunikasi (JTIK)*, vol. 16, no. 1, pp. 6–13, Mar. 2025.
- [2] D. P. Ardiansyah, A. S. Y. Irawan, and E. H. Nurkifli, "Rancang bangun sistem pelayanan laundry berbasis website menggunakan ReactJS," *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, vol. 12, no. 3, pp. 2572–2581, Aug. 2024.
- [3] E. K. Hazanah, A. Oktaviani, and R. Nurfalah, "Penerapan model waterfall dalam perancangan sistem informasi berbasis web pada Devan Petshop," *Jurnal Responsif*, vol. 4, no. 2, pp. 135–141, Aug. 2022.
- [4] K. C. Laudon and J. P. Laudon, *Management Information Systems: Managing the Digital Firm*, 17th ed. Hoboken, NJ: Pearson, 2022.
- [5] N. Aliah, A. Suwarni, and N. Natsir, "Evaluating the impact of innovative assignment design on student engagement and learning outcomes in higher education," *Indonesian Journal of English Language, Literature, and Culture*, vol. 6, no. 2, pp. 78–90, 2022.
- [6] S. P. Dewi and R. A. Pratama, "System development life cycle models: A comparative study and application," *Journal of Information Systems Engineering and Business*, vol. 7, no. 1, pp. 14–22, Feb. 2023.
- [7] A. M. Hidayat and D. Sari, "Analysis of waterfall model application in software development projects," *Journal of Computer Science and Technology*, vol. 10, no. 2, pp. 55–63, Jun. 2022.
- [8] M. Kurniawan and L. A. Prasetyo, "Application of UML in software development: A case study," *International Journal of Software Engineering and Applications*, vol. 12, no. 3, pp. 34–42, May 2023.

- [9] P. Yenti and G. Farell, "Rancang bangun sistem informasi pengelolaan tugas mahasiswa pada matakuliah pemrograman sistem bergerak berbasis web," *Jurnal Vocational Teknik Elektronika dan Informatika*, vol. 9, no. 1, pp. 1–10, Mar. 2021.
- [10] W. Erawati, S. Heristian, and R. A. Purnama, "Rancang bangun sistem informasi akademik berbasis website dengan metode SDLC," *Computer Science (CO-SCIENCE)*, vol. 3, no. 2, pp. 68–75, Jul. 2023.
- [11] R. Yani, M. Mansur, and K. Kasmawi, "Sistem informasi penjadwalan les privat menggunakan framework CodeIgniter," *Information System for Educators and Professionals*, vol. 5, no. 1, pp. 11–20, Dec. 2020.
- [12] M. Weidt *et al.*, "Advances in automated support for requirements engineering: a systematic literature review," *Req. Eng.*, vol. 28, pp. 1–38, 2023.
- [13] Z. N. Muhammad, A. Meiriza, P. Putra, N. R. Oktadini, and P. E. Sevtiyuni, "Perancangan sistem informasi manajemen aset laboratorium berdasarkan pendekatan human centered design (HCD)," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 3, no. 6, Jun. 2023.
- [14] M. Syarif and W. Nugraha, "Pemodelan diagram UML sistem pembayaran tunai pada transaksi e-commerce," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 4, no. 1, p. 64, Jan. 2020.
- [15] Hendy, "Pemodelan sistem menggunakan UML (Unified Modelling Language)," Conference Paper, Jul. 2019.
- [16] S. Al-Fedaghi, "Classes in object-oriented modeling (UML): further understanding and abstraction," *arXiv*, Jun. 2021.
- [17] M. A. Adeagbo, J. E. T. Akinsola, A. A. Awoseyi, and F. Kasali, "Project implementation decision using software development life cycle models: a comparative approach," *J. Comput. Sci. Its Appl.*, vol. 28, no. 1, 2021.
- [18] M. Zen, I. Irwan, H. Hafni, and M. D. P. Ananda, "Implementasi dan pengujian menggunakan metode BlackBox Testing pada sistem informasi tracer study," *Bull. Comput. Sci. Res.*, vol. 4, no. 4, Apr. 2024.
- [19] A. W. Putri, M. R. H., and N. C. I. B., "Evaluasi pengujian penerimaan pengguna (user acceptance testing) pada sistem informasi akademik Universitas Teknologi AKBA Makassar," *Inventor: J. Innov. Technol. Educ. Inf.*, vol. 3, no. 2, 2024.