belantika Pendidikan
ISSN 2720-9601
Vol.7 (2), November 2024, pp. 73-83

Task Scheduler Untuk Optimalisasi Laporan Pertanggungjawaban Keuangan di Perguruan Tinggi

Nuribut Setyawati*, Gres Dyah Kusuma Ningrum*, Febrianto Alqodri*
* Fakultas Teknik, Universitas Negeri Malang

INFO ARTIKEL

Riwayat Artikel:

Diterima: Tgl-Bln-Thn
Disetujui: Tgl-Bln-Thn

Kata kunci:

Task Scheduler
Optimamalisasi
Laporan keuangan

ABSTRAK

Abstract: At the end of the year, there is an abundance of reports because students tend to collect reports after the finance officer's deadline for reporting accountability (LPJ) for student activities. By waterfall approach with the task scheduler feature, the LPJ FT UM application was created. It can handle several aspects of LPJ and also reminding students to submit activity reports quickly. This tool streamlines and reduces the amount of time students need to handle LPJ operationally. Monitoring the status of LPJs sent by students will be simple for finance officers. Based on operational test this apps reach 93%, so it's very worthy to use.

Abstrak: Keterlambatan pelaporan pertanggungjawaban (LPJ) kegiatan kemahasiswaan mengakibatkan penumpukan laporan di akhir tahun, hal ini dikarenakan kecenderungan mahasiswa mengumpulkan melebihi batas waktu yang telah ditentukan oleh petugas keuangan. Aplikasi LPJ FT UM dikembangkan menggunakan metode waterfall diintegrasikan dengan fitur *task scheduler* yang bisa melakukan manajemen LPJ seperti proses pengunggahan, verifikasi, serta selalu memberikan pengingat berkala bagi mahasiswa untuk segera menyerahkan laporan kegiatan. Secara operasional, aplikasi ini mempermudah serta mempersingkat waktu yang dibutuhkan mahasiswa dalam mengurus LPJ. Bagi petugas keuangan akan mudah melakukan monitoring status LPJ yang dikirimkan oleh mahasiswa. Berdasarkan hasil uji coba operasional, aplikasi ini memiliki mencapai 93%, yang artinya aplikasi ini sangat layak digunakan.

Alamat Korespondensi:

Nuribut Setyawati
Fakultas Teknik
Universitas Negeri Malang
Jl. Semarang No. 5 Malang
E-mail: nuribut.setyawati@um.ac.id

PENDAHULUAN

Di era modern ini, pengelolaan keuangan dalam organisasi pendidikan menjadi semakin kompleks, terutama dalam konteks kegiatan kemahasiswaan. Di Fakultas Teknik Universitas Negeri Malang (UM) yang merupakan instansi PTN-BH (Perguruan Tinggi Negeri Berbadan Hukum), mahasiswa terlibat dalam berbagai aktivitas yang memerlukan pendanaan dan pertanggungjawaban yang sesuai. Namun, sering kali timbul masalah seperti keterlambatan dalam pelaporan yang mengakibatkan penumpukan berkas dan lambannya proses administrasi. Hal ini mengakibatkan terjadinya penumpukan laporan kegiatan pada akhir tahun karena mahasiswa melakukan laporan secara mendadak dan tidak terstruktur secara baik karena masih menggunakan laporan kertas/biasa.

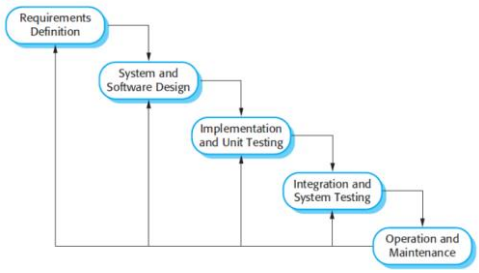
Untuk mengatasi masalah ini, diperlukan solusi yang dapat menyederhanakan proses pelaporan dan memastikan mahasiswa dapat tertib dengan jadwal yang telah ditetapkan oleh tim keuangan. Salah satu pendekatan yang dapat diterapkan adalah pengembangan aplikasi berbasis teknologi informasi yang dirancang khusus untuk kebutuhan ini.

Aplikasi yang dilengkapi dengan fitur *task scheduler* yang dapat membantu pengelolaan pelaporan keuangan dengan lebih terstruktur dan tepat waktu (Moedjahedy, 2018). Dengan sistem yang lebih baik, diharapkan dapat meningkatkan efisiensi dan efektivitas proses administrasi. Aplikasi LPJ FT UM dikembangkan dengan menggunakan metode pengembangan *waterfall*. Fitur utama aplikasi ini mencakup *task scheduler* (Huang, Xie , An, Su, & Zhang, 2024) yang memudahkan penjadwalan dan pengawasan pelaporan. Selain itu, aplikasi ini memungkinkan komunikasi online antara mahasiswa dan tim keuangan, yang mengurangi kebutuhan untuk pertemuan tatap muka dan mempermudah proses monitoring (Elfindri, 2011). Implementasi aplikasi LPJ FT UM diharapkan dapat memberikan manfaat signifikan bagi semua pihak yang terlibat dalam pengelolaan kegiatan kemahasiswaan. Dengan adanya sistem yang terintegrasi dan otomatis, diharapkan masalah keterlambatan pelaporan dapat diminimalisir, dan proses administrasi menjadi lebih lancar dan efisien.

METODE

Metode pengembangan dari aplikasi ini adalah metode pengembangan waterfall (Sommerville, 2001). Pemilihan metode ini dikarenakan alur kerja yang sekuensial dan terdapat evaluasi di akhir pengembangan (Gurung, Shah, & Jaiswal, 2020). Adapun langkah yang dilakukan setiap tahap adalah

1. Requirement Definition : memberikan informasi terkait kebutuhan fitur dan batasan pengembangan. Pada tahap ini biasanya akan menghasilkan *Software Requirement Spesification (SRS)* (Baskoro, Andrahsmara, Darnoto, & Tofan, 2021).
2. System and Software Design: melakukan perancangan aplikasi berupa pengembangan Data Flow Diagram (DFD) (Surono, 2014), flow chart, Unified Manipulation Language (UML) (Sonata & Vina, 2019), dan Entity Relational Diagram (ERD) (Anggit & Hartanto, 2012).
3. Implementation and Unit Testing: melakukan implementasi desain sistem serta melakukan uji coba terbatas terhadap modul-modul aplikasi yang dikembangkan menggunakan metode pengujian *black-box testing* tipe *Boundary Value Analysis* (Rayhan, 2024).
4. Integration and System Testing : melakukan instalasi aplikasi ke server dengan integrasi *existing system*.
5. Operation and Maintenance: penggunaan aplikasi secara operasional dan perawatan (jika diperlukan) serta emantauan performa aplikasi secara berkala untuk memastikan aplikasi berjalan dengan baik.



Gambar 1. Metode Pengembangan *Waterfall*

Requirement Definition

Pada tahap *Requirement Definition* atau analisis kebutuhan, dibuat suatu *test case* yang berfungsi sebagai tujuana akhir dari produk yang dikembangkan. Uraian kebutuhan didasarkan apda kebutuhan tim keuangan yang bertugas untuk melakukan pengelolaan/pengumpulan laporan pertanggungjawaban (LPJ) dari kegiatan mahasiswa. Secara spesifik, aplikasi harus mampu berjalan dengan kebutuhan umum dan khusus untuk mempermudah pengelolaan LPJ. Nilai dari setiap kebutuhan akan direpresentasikan dalam matriks skala Likert dengan lima pilihan dan di analisis hasil akhirnya.

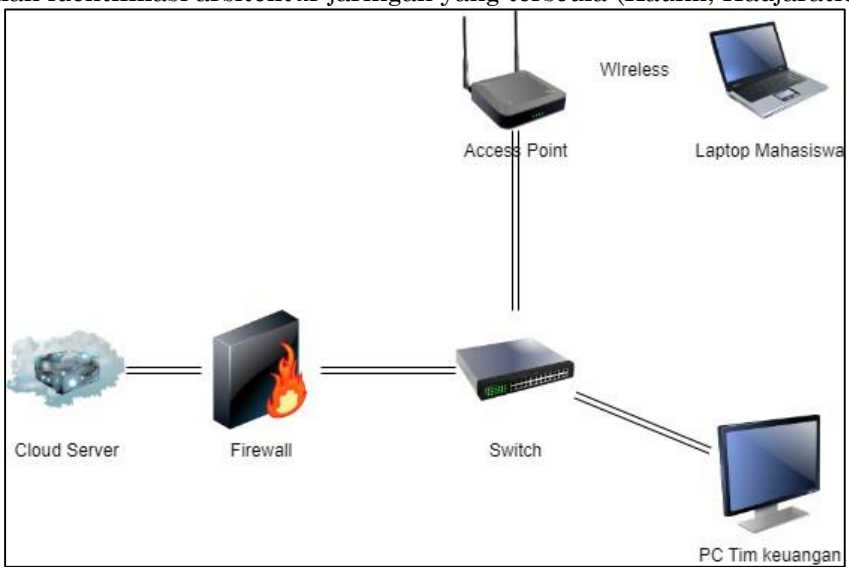
Tabel 1. Analisis Kebutuhan Aplikasi LPJ

Aspek Umum	**Parameter/Luaran**
Aplikasi mampu berjalan dengan platform website	Likert skala 5
Aplikasi cepat dan mudah digunakan	Likert skala 5

Aspek Umum	**Parameter/Luaran**
Aplikasi dapat mengurangi kesalahan yang terjadi ketika di review	Likert skala 5
Aplikasi tidak terdapat error	Likert skala 5
Aspek Khusus	
Fitur *task scheduler* tepat waktu untuk menjalankan reminder	Likert skala 5
Fitur *task scheduler* tepat kepada sasaran mahasiswa	Likert skala 5
Stakeholder dapat dengan mudah monitoring status LPJ	Likert skala 5

System and Software Design

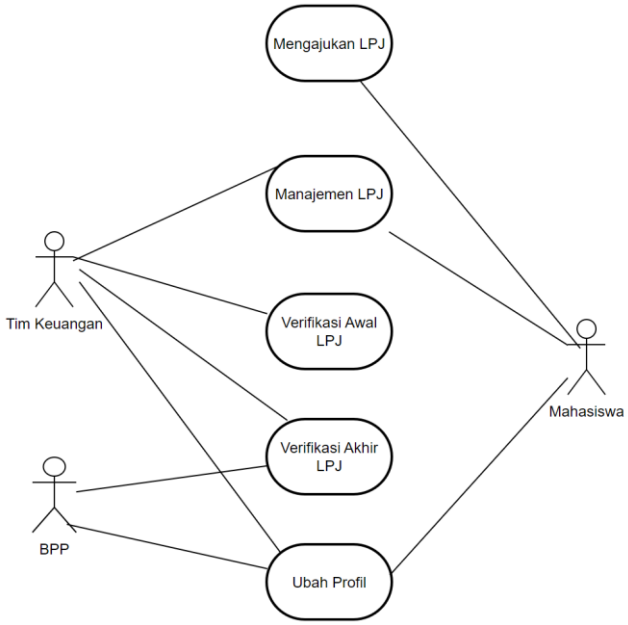
Tahap *System and Software Design* berfokus pada pengembangan desain sistem dan rencana implementasi ke dalam baris kode/arsitektur/skema. Identifikasi *existing* arsitektur dilakukan untuk melihat kesiapan lingkungan pengembangan untuk memasukan sistem informasi atau aplikasi ini dijalankan. Gambar 2 menunjukkan identifikasi arsitektur jaringan yang tersedia (Kadim, Hadjaratie, & Muthia, 2023).



Gambar 2. *Existing* Arsitektur Jaringan

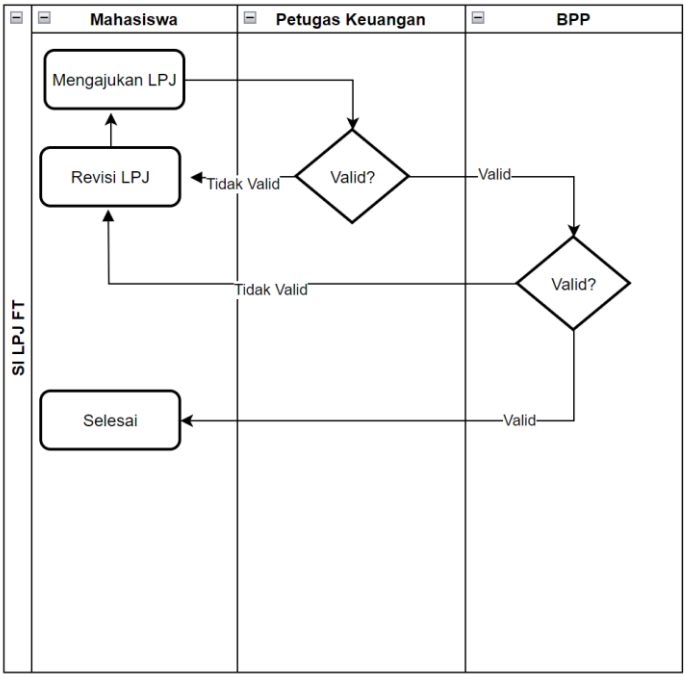
Jika ditelaah lebih detail, aplikasi yang dikembangkan akan diinstal pada *cloud server* yang nantinya diakses oleh tim keuangan dan mahasiswa. Berdasarkan identifikasi yang ada, *existing cloud server* menggunakan sistem informasi Linux (Rahayu & Sensuse, 2017) dan memiliki plugin website Hypertext Processor (PHP) dan basis data MariaDB. Aplikasi dikembangkan menggunakan bahasa pemrograman PHP (Group, 2024) yang terintegrasi dengan basisdata MariaDB (Foundation, 2024). Penggunaan framework PHP Yii2 (Syamsudin, 2021) dipilih karena menggunakan alur framework *Model View Controller* (MVC) dan mampu menangani pengolahan database yang banyak.

Secara sistem, desain *UML* yang dikembangkan menggunakan representasi *use case diagram* yang merepresentasikan aktor dan hak akses yang dimilikinya di sistem. Hal ini akan mempermudah pengembang untuk mendapatkan intisari kebutuhan dan hak setiap pengguna di sistem. Gambar 3 menunjukkan diagram hak akses pengguna.



Gambar 3. *Use Case Diagram* Aplikasi LPJ

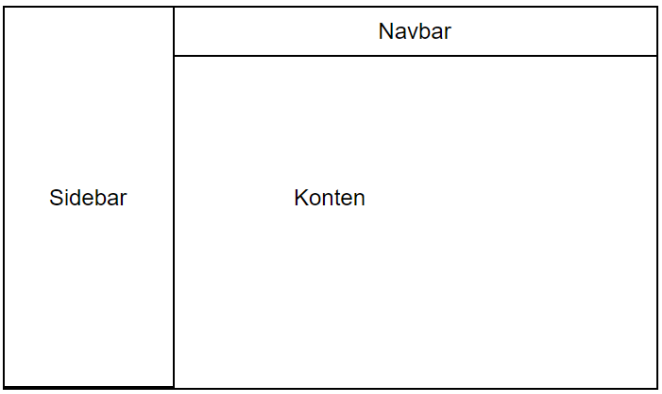
Alur proses pelaksanaan pengajuan hingga verifikasi LPJ dilakukan berdasarkan alur proses yang ditunjukkan oleh Gambar 4. Setiap langkah/alur harus diikuti sesuai prosedur yang berlaku agar tidak ada yang terlewat. *Activity diagram* sesuai Gambar 4 dibuat untuk mempermudah dalam implementasi baris kode program.



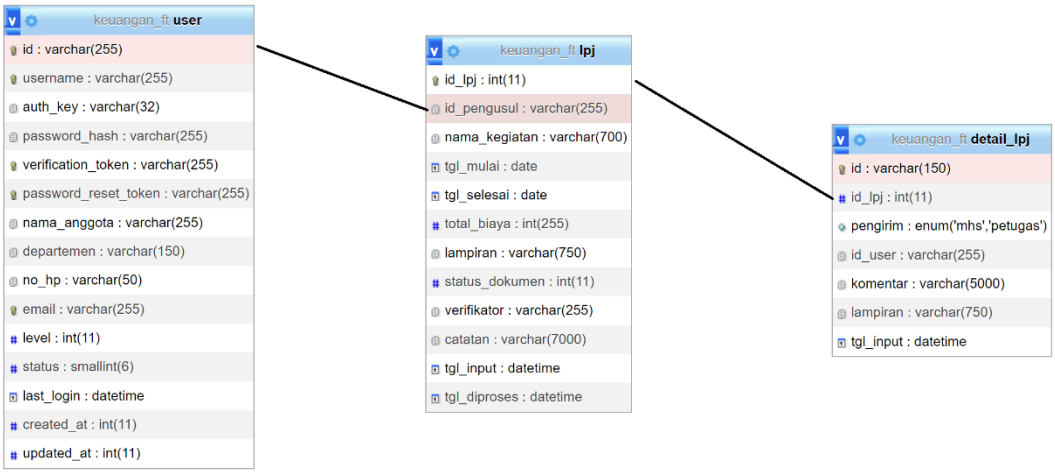
Gambar 4. *Activity Case Diagram* Aplikasi LPJ

Tampilan aplikasi berupa *user interface* di desain menggunakan rancangan desain di platform website. Hal ini dikarenakan kecenderungan pengguna menggunakan platform ini untuk mengakses, melihat, dan merevisi data yang akan dikirimkan pada aplikasi. Adapun tampilan *user interface* aplikasi ditunjukkan

oleh Gambar 5. Penyimpanan data dilakukan kepada *database*/basis data yang nantinya disimpan di server. Penyimpanan dilakukan sesuai prosedur penyimpanan data yang telah berjalan sesuai penyedia database. Representasi desain database yang digunakan ditunjukkan oleh ERD yang ditunjukkan oleh Gambar 6.



Gambar 5. *User Interface* Aplikasi LPJ



Gambar 6. Entity Relationship Diagram (ERD) Aplikasi LPJ

Implementation and Unit Testing

Implementation and Unit Testing memiliki fokus untuk melakukan pengkodean ke aplikasi editor, melakukan integrase *plugin* yang diperlukan dan melakukan uji coba terhadap modul-modul yang dikembangkan berdasarkan kebutuhan yang telah ditentukan. Untuk uji coba modul aplikasi/*unit testing,* dilakukan beberapa skenario uji coba berdasarkan *test case* yang telah ditentukan. Adapun instrument uji coba dari implementasi diadaptasi dari penilaian perangkat lunak seperti yang ditunjukkan oleh Tabel 2 (Wahono, 2006). Uji coba menggunakan skenario *black-box boundary value analysis* yang berfokus pada *input* dan *output* yang dilakukan pada aplikasi. Subjek coba pada testing ini kepada pengembang aplikasi/sistem informasi. Jika uji coba yang dilakukan pada aplikasi sesuai, maka akan bernilai sesuai/benar/*true.* Revisi dilakukan jika nilai persentase kesesuaian tidak 100% atau berarti ada fungsi yang tidak sesuai. Prinsip pareto tidak digunakan pada pengujian ini agar meminimalisir toleransi kesalahan yang mungkin dihasilkan oleh aplikasi (Muda, Kleden, & Br. Ginting, 2020).

Tabel 2. Item Uji Unit Testing

Aspek Umum	Parameter/Luaran
Efektif dan Efisien	True/False
Reliabilitas	True/False
Maintainabilitas	True/False
Ketepatan Pemilihan Jenis Aplikasi/Software/Tool	True/False
Kompatibilitas	True/False
Reusabilitas	True/False

Integration and System Testing

Tahap ini berfokus pada integrasi yang dilakukan setelah aplikasi berjalan dan dilakukan instalasi pada arsitektur jaringan yang sebenarnya. Uji coba dilakukan kepada pengguna aplikasi berdasarkan instrumen yang telah dituliskan pada Tabel 1. Jika tingkat persentase uji coba diatas 76%, maka aplikasi termasuk dalam kriteria sangat layak untuk digunakan (Yulistina & Dwi Arianti, 2019).

Operation and Maintenance

Tahap ini merupakan tahap akhir dari proses pengembangan, dimana aplikasi akan digunakan dan bisa diakses secara intranet. Penggunaan intranet dipilih karena lebih aman dan semua fitur yang diintegrasikan pada aplikasi bisa berjalan secara normal. Pemeliharaan kepada aplikasi bisa dilakukan jika terdapat *bug* atau *error* yang terjadi ketika proses operasional dilakukan (Suteja & Munggaran, 2020).

HASIL

Berdasarkan metode yang telah dipilih dan dijelaskan pada tahap sebelumnya, aplikasi berhasil dikembangkan dan bisa berjalan secara operasional. Terdapat tiga aktor utama dalam penggunaan aplikasi, yaitu mahasiswa, petugas keuangan, dan petugas Bendahara Pengeluaran Pembantu (BPP). Adapun hasil/luaran dari setiap langkah atau tahap metode yang dipilih ditunjukkan oleh Tabel 3. Hasil uji coba terhadap tiga stakeholder didapatkan data bahwa aplikasi dapat berjalan 100%.

Tabel 3. Hasil Pengembangan dengan Metode *Waterfall*

Tahap Pengembangan	Luaran	Dokumentasi Luaran
Requirements Definition	Analisis kebutuhan perangkat lunak	*Software Requirement Specification (SRS)* (Serafintino & Susilowati, 2022)
System and Software Design	Desain alur system, antarmuka, dan basis data	UML, DFD, UML, UI/UX, dan ERD
Implementation and Unit Testing	*Unit test case*	*Black-box result*
Integration and System Testing	*System case*	*Black-box result*
Operation and Maintenance	Operasional	Buku manual penggunaan dan catatan *maintenance*

Berdasarkan Tabel 2 terkait tahap implementasi dan unit testing yang bertujuan untuk melakukan uji coba terhadap modul aplikasi yang dikembangkan. Uji coba dilakukan oleh pengembang aplikasi yang memiliki pengalaman pengembangan aplikasi. Adapun hasil uji coba ditunjukkan oleh Tabel 3.

Tabel 3. Hasil Uji Coba Unit Testing

Aspek Umum	Parameter/Luaran
Efektif dan Efisien	100%
Reliabilitas	100%
Maintainabilitas	100%
Ketepatan Pemilihan Jenis Aplikasi/Software/Tool	100%

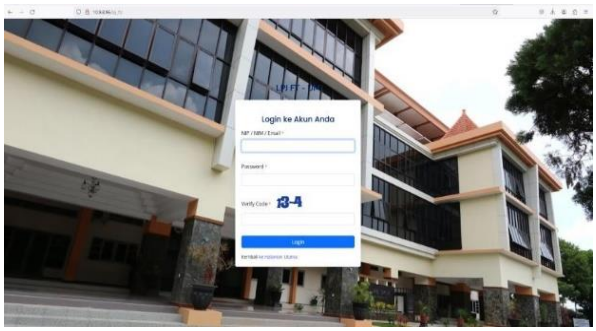
Aspek Umum	Parameter/Luaran
Kompatibilitas	100%
Reusabilitas	100%

Tabel 1 yang menunjukkan terkait kebutuhan terhadap aplikasi yang telah dikembangkan. Terdapat 3 kategori subjek coba yang menggunakan aplikasi dan semua dilakukan uji coba terkait proses operasional aplikasi berdasarkan kebutuhan yang telah didefisinikan. Subjek coba mahasiswa diperoleh dari perwakilan mahasiswa yang aktif berorganisasi di lingkup fakultas, petugas keuangan merupakan penanggung jawab administrative LPJ, dan petugas BPP merupakan bendahara tingkat fakultas. Adapun hasil uji coba penggunaan dalam representasi persentase ditunjukkan oleh Tabel 4.

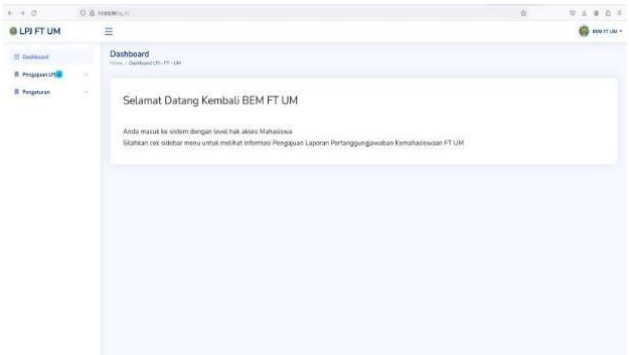
Tabel 4. Hasil Uji Coba Operasional kepada Subjek Coba

Aspek Umum	Mahasiswa	Petugas Keuangan	BPP
Aplikasi mampu berjalan dengan platform website	97%	95%	92%
Aplikasi cepat dan mudah digunakan	95%	98%	90%
Aplikasi dapat mengurangi kesalahan yang terjadi ketika di review	94%	95%	92%
Aplikasi tidak terdapat error	89%	90%	90%
Aspek Khusus			
Fitur *task scheduler* tepat waktu untuk menjalankan reminder	95%	90%	90%
Fitur *task scheduler* tepat kepada sasaran mahasiswa	97%	96%	94%
Stakeholder dapat dengan mudah monitoring status LPJ	95%	94%	92%

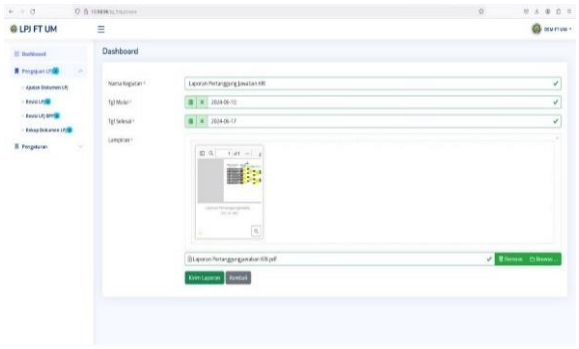
Aplikasi dikembangkan menggunakan platform website. Gambar 7 menunjukkan tampilan login aplikasi. Mahasiswa akan melakukan unggah laporan kegiatan dan memerlukan verifikasi oleh petugas keuangan. Gambar 8 menunjukkan tampilan akun mahasiswa dan Gambar 9 menunjukkan form pengunggahan laporan pertanggungjawaban (LPJ).



Gambar 7. Tampilan Login Aplikasi LPJ FT UM

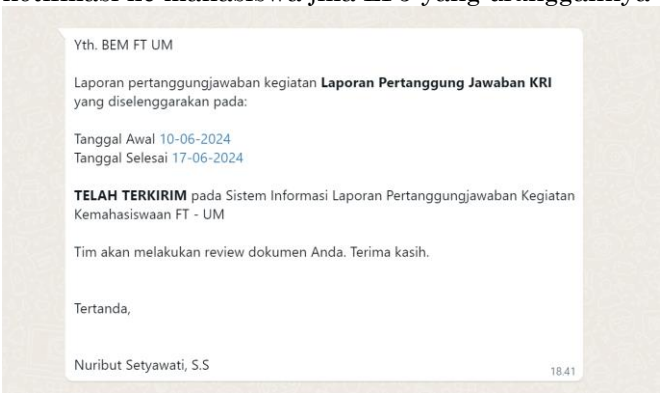


Gambar 8. Tampilan Dashboard Mahasiswa

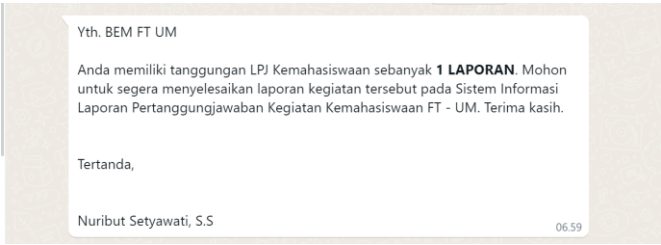


Gambar 9. Tampilan Form Unggah Laporan di Mahasiswa

Data yang telah masuk akan men-*trigger* notifikasi menggunakan platform WhatsApp ke HP mahasiswa untuk memastikan bahwa data LPJ telah terunggah. Gambar 10 menunjukkan notifikasi WhatsApp di HP mahasiswa. Petugas keuangan dapat melakukan verifikasi terhadap LPJ yang telah digunggah seperti ditunjukkan oleh Gambar 11. Mahasiswa akan mendapatkan notifikasi setiap hari melalui WhatsApp jika status LPJ yang diunggah belum dinyatakan valid oleh BPP seperti ditunjukkan oleh Gambar 12. Laporan pertanggungjawaban yang telah diverifikasi petugas keuangan dan BPP dan dinyatakan valid, maka secara otomatis akan mengirimkan notifikasi ke mahasiswa jika LPJ yang diunggahnya valid.



Gambar 10. Notifikasi WhatsApp Unggah Laporan



Gambar 11. Proses Reminder Kegiatan

Gambar 12. Notifikasi Jika LPJ Valid

PEMBAHASAN

Secara operasional aplikasi laporan pertanggungjawaban (LPJ) FT UM bisa berjalan melalui akses website. Penggunaan website dipilih dikarenakan seluruh stakeholder memiliki penguasaan terhadap platform tersebut. Berdasarkan Gambar 2, penggunaan server lokal dipilih dikarenakan aplikasi ini berhubungan erat dengan sensitifitas lembaga, yaitu berhubungan dengan keuangan. Hal ini akan mencegah penyebabnya informasi keuangan kepada pihak yang tidak berkepentingan. Melalui aplikasi ini mahasiswa bisa mengajukan LPJ secara intranet dan realtime. Fitur *task scheduler* yang terintegrasi pada aplikasi akan memastikan bahwa setiap unggahan data yang dikirim oleh mahasiswa selalu dicek secara berkala.

Pada tahap pengembangan, *use case diagram* yang telah dirancang sesuai Gambar 3 diimplementasikan sesuai alur kerja yang telah ditunjukkan oleh Gambar 4. Implementasi ini memungkinkan terjadinya transfer data antara mahasiswa, petugas keuangan, dan petugas BPP. Pada Tabel 3 didapatkan bahwa metode yang digunakan merupakan metode yang runtut dan memastikan setiap tahapan telah selesai dilaksanakan. Hal ini akan memastikan bahwa kebutuhan yang direncanakan telah terimplementasi dengan baik. Pada langkah pertama, analisis kebutuhan didefinisikan bahwa 1) mahasiswa harus bisa mengunggah LPJ, 2) petugas keuangan bisa melakukan verifikasi, dan 3) petugas BPP bisa mendapatkan informasi guna memverifikasi LPJ yang diunggah. Proses desain dan implementasi yang dilakukan terhadap aplikasi, menghasilkan bahwa aplikasi dapat bekerja 100% (berdasarkan Tabel 3) sesuai hasil uji pengembang aplikasi. Jika dilihat dari sudut pandang operasional, aplikasi memiliki tingkat kelayakan dengan rata-rata 93%, hasil ini mengindikasikan bahwa aplikasi termasuk dalam kategori "Sangat Layak" untuk digunakan karena sudah menyesuaikan dengan kebutuhan setiap stakeholder. Hal tersebut mengindikasikan bahwa mahasiswa bisa melakukan unggah laporan pertanggungjawaban, petugas keuangan bisa memverifikasi data keuangan, dan petugas BPP bisa memvalidasi laporan tersebut.

Urutan penggunaan aplikasi dimulai dari mahasiswa yang mengakses portal seperti yang ditunjukkan oleh Gambar 8, kemudian melakukan unggah laporan seperti Gambar 9. Jika data laporan telah terunggah maka mahasiswa akan mendapatkan notifikasi melalui platform WhatsApp seperti yang ditunjukkan oleh Gambar 10. Jika mahasiswa tidak segera melakukan proses untuk segera menyelesaikan laporan pertanggungjawabannya, maka secara otomatis sistem akan memberikan notifikasi setiap hari agar mahasiswa segera menyelesaikan laporan angkatan tersebut. Gambar 11 merupakan contoh pengingat berdasarkan task scheduler yang diintegrasikan dalam aplikasi guna memberikan informasi bagi mahasiswa untuk segera memproses laporan pertanggungjawabannya. Jika laporan telah diproses oleh petugas keuangan dan juga petugas BPP maka mahasiswa akan mendapatkan notifikasi WhatsApp berupa informasi terkait laporan pertanggungjawabannya yang menyatakan laporan pertanggungjawaban tersebut valid. Jika laporan pertanggungjawaban telah dinyatakan valid, maka mahasiswa tidak akan mendapatkan pengingat bahwa laporannya sedang diproses atau perlu segera diselesaikan.

Melalui aplikasi ini, diharapkan manajemen pengelolaan LPJ bisa lebih optimal. Kebijakan terkait pengumpulan laporan pertanggungjawaban juga bisa lebih teratur dan sesuai dengan rentang waktu yang telah ditetapkan. Aplikasi memiliki keterbatasan hanya dalam pengelolaan laporan kegiatan saja,

kedepannya jika memungkinkan aplikasi bisa dikembangkan mulai proses pengajuan maupun verifikasi kegiatan, sehingga alur proses mulai kegiatan direncanakan, di verifikasi hingga proses pelaporan lebih mudah untuk dikelola.

SIMPULAN

Berdasarkan latar belakang, metodologi pengembangan, dan hasil implementasi, dapat disimpulkan bahwa aplikasi laporan pertanggungjawaban (LPJ) Fakultas Teknik, Universitas Malang (FT UM) berfungsi dengan optimal, memenuhi seluruh kebutuhan yang telah didefinisikan. Aplikasi ini berjalan 100% sesuai harapan, mempermudah serta mempercepat proses pengurusan laporan pertanggungjawaban mahasiswa. Secara operasional, mahasiswa dapat mengajukan LPJ secara intranet dan *real-time* tanpa perlu bertatap muka, sementara laporan kegiatan terjadwal dan terpantau mulai dari pengajuan proposal, pencairan dana, hingga pembuatan laporan pertanggungjawaban. Fitur *task scheduler* yang diintegrasikan memastikan data yang diunggah diperiksa secara berkala, dan sistem memberikan notifikasi otomatis kepada mahasiswa melalui WhatsApp untuk mengingatkan mereka tentang status laporan mereka. Tim keuangan dan petugas BPP dapat memantau, memverifikasi, dan memvalidasi laporan secara online sesuai dengan jadwal yang telah direncanakan, memastikan proses yang efisien dan transparan. Berdasarkan hasil uji coba dari pengembang aplikasi, aplikasi dapat berjalan 100%, sedangkan untuk uji coba operasional aplikasi memiliki tingkat persentase 93% yang artinya termasuk dalam kriteria “Sangat Layak”.

DAFTAR RUJUKAN

Anggit, D. U., & Hartanto. (2012). *Sistem Basis Data menggunakan Microsoft SQL Server 2005.* Yogyakarta: Penerbit Andi.

Baskoro, F., Andrahsmara, R. A., Darnoto, B. R., & Tofan, Y. A. (2021). A Systematic Comparison of Software Requirements Classification. *IPTEK The Journal for Technology and Science, 32*(3), 184-193. doi:10.12962/j20882033.v32i3.13005

Elfindri. (2011). Beberapa Teknik Monitoring dan Evaluasi. *Jurnal Kesehatan Komunitas, 1*(3), 106-128.

Foundation, M. (2024, August 14). *Download MariaDB Server*. Diambil kembali dari MariaDB Server: https://mariadb.org/download/?t=mariadb&p=mariadb&r=10.5.26&os=windows&cpu=x86_64&pkg=msi&mirror=vhost_sg

Group, P. (2024, Sept 26). *PHP: Hypertext Preprocessor*. Diambil kembali dari PHP: Hypertext Preprocessor: https://www.php.net/

Gurung, G., Shah, R., & Jaiswal, D. (2020). Software Development Life Cycle Models-A Comparative Study. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, *6*(4), 30-37. doi:10.32628/CSEIT206410

Huang, X., Xie , M., An, D., Su, S., & Zhang, Z. (2024). Task scheduling in cloud computing based on grey wolf optimization with a new encoding mechanism. *Parallel Computing, 122*, 1-9. doi:https://doi.org/10.1016/j.parco.2024.103111

Kadim, A. A., Hadjaratie, L., & Muthia, M. (2023). Implementasi Framework Laravel Dalam Pembuatan Sistem Pencatatan Notula Berbasis Website. *Jurnal Sistem Informasi Bisnis, 13*(1), 45-51. doi:https://doi.org/10.21456/vol13iss1pp45-51

Moedjahedy, J. (2018). Implementasi Cron Job Linux Sebagai Bel Pergantian Kelas Otomatis Di Universitas Klabat. *CogITo Smart Journal, 4*(1), 1-10.

Muda, M. F., Kleden, M., & Br. Ginting, K. (2020). Estimasi Parameter Distribusi Pareto. *Jurnal Diferensial, 2*(1), 1-7.

Rahayu, P., & Sensuse, D. I. (2017). Penilaian Implementasi e-Government di PUSTEKOM Kemendikbud berbasis metode PEGI. *7*(2), 139-145. doi:10.21456/vol7iss2pp139-145

Rayhan, P. F. (2024). Teknik Boundary Value Analysis pada Blackbox Testing untuk Aplikasi Buku Catatan Harian. *REPOSITOR, 6*(1), 69-78.

Serafintino, R. A., & Susilowati, M. (2022). DOKUMEN SOFTWARE REQUIREMENT SPESIFICATION (SRS) SISTEM INFORMASI PEMASARAN USAHA JASA PERCETAKAN DAN IKLAN. *Kurawal - Jurnal Teknologi, Informasi Dan Industri, 5*(2), 117-128. doi:https://doi.org/https://doi.org/10.33479/kurawal.v5i2.646

Sommerville, I. (2001). *Software Engineering Edisi 6.* Jakarta: Erlangga.

Sonata, F., & Vina, W. S. (2019). Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer. *Jurnal Komunika, 8*(1), 22-31. doi:10.31504/komunika.v8i1.1832

Surono. (2014). Data Flow Diagram (DFD) Pada Apotek Candra Kota Jambi. *Jurnal Ilmiah Universitas Batang hari Jambi, 14*(4).

Suteja, E. R., & Munggaran, L. C. (2020). Penerapan Pemeliharaan Systems Maintenance Life Cycle Bank Swasta Nasional Berdasarkan IEEE dan ISO. *Journal of Information System, Informatics and Computing, 4*(1), 46-54.

Syamsudin, E. (2021). Analysis of Information Systems Inpassing Functional Positions of Engineers and Engineers on-line. *INTERNATIONAL JOURNAL OF MULTI SCIENCE, 2*(1), 28-43.

Wahono, R. S. (2006, June 23). *romisatriawahono.net*. Diambil kembali dari Aspek Rekayasa Perangkat Lunak dalam Media Pembelajaran: https://romisatriawahono.net/2006/06/23/media-pembelajaran-dalam-aspek-rekayasa-perangkat-lunak/

Yulistina, D., & Dwi Arianti, B. D. (2019). E-Katalog Sebagai Sistem Informasi Pemasaran Kopi Sapit Berbasis Web. *Edumatic: Jurnal Pendidikan Informatika, 3*(2), 45-52.