

Pengembangan Aplikasi Manajemen Litabmas Pendanaan Mandiri dengan Pendekatan Hierarchical Model-View-Controller (HMVC)

Eka Dyar Wahyuni^{1*}, Mohamad Irwan Afandi², Agung Brastama Putra³

^{1,2,3} Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur
Jl. Rungkut Madya No.1, Gn. Anyar, Kec. Gn. Anyar, Surabaya, Jawa Timur, Indonesia 60294

e-mail: ¹ekawahyuni.si@upnjatim.ac.id, ²mohamadafandi.si@upnjatim.ac.id, ³agungbp.si@upnjatim.ac.id

(*) Corresponding Author

Artikel Info : Diterima : 21-11-2023 | Direvisi : 23-01-2024 | Disetujui : 30-01-2024

Abstrak - Pengelolaan pelaksanaan kegiatan litabmas di suatu perguruan tinggi merupakan suatu kegiatan yang cukup kompleks dan rentan menghadapi permasalahan, apalagi jika dikelola secara manual melalui penyerahan dokumen secara fisik. Arsip sering hilang atau bahkan rusak, kesulitan dalam melacak tanggungan luaran litabmas (artikel, jurnal, paten/hki dll) periode sebelumnya hingga kesulitan menyediakan informasi dengan cepat. Artikel ini membahas pengembangan aplikasi manajemen litabmas pendanaan mandiri mempergunakan metode *Waterfall*. Pada saat pengkodean, arsitektur kode dibangun dengan mempergunakan pendekatan *Hierarchical Model-View-Controller* (HMVC). Pendekatan ini meminimalkan ketergantungan antar modul sehingga memudahkan proses pengembangan aplikasi secara tim. Aplikasi ini dibuat untuk memudahkan pengguna dalam mengelola seluruh siklus pelaksanaan kegiatan litabmas, mulai dari pembuatan data master litabmas, pembukaan periode litabmas, pengajuan proposal, evaluasi, hingga pelaporan hasil dan luaran litabmas. Hasil akhir dari penelitian ini adalah, aplikasi telah diuji dengan metode *blackbox*, dan secara keseluruhan menghasilkan 23 *usecase* dari 6 level pengguna - admin, pimpinan, peneliti, reviewer, auditor dan pengunjung umum - yang sudah 100% memberikan *output* sesuai yang diharapkan.

Kata Kunci : pengembangan aplikasi, aplikasi manajemen litabmas, HMVC, CI

Abstracts - In higher education, overseeing the execution of litabmas activities is a challenging and error-prone task, particularly when done manually through the physical submission of documents. Information is not always readily available, archives are frequently lost or even damaged, and tracing Litabmas external responsibility (articles, journals, patents/IPRs, etc.) from earlier periods can be challenging. This article describes how the *Waterfall* approach was used to construct the Litabmas management application. The *Hierarchical Model-View-Controller* (HMVC) technique is used to build the code architecture during the coding process. By reducing intermodular dependencies, this method makes it easier for a team to design an application. This application was developed to facilitate the management of the full cycle of Litabmas activity implementation for users, from generating Litabmas master data to initiating the Litabmas period, submitting proposals, conducting evaluations, and reporting Litabmas outputs and outcomes. The application has been tested using the *black box* method, and as a result, 23 use cases from six levels of users —administrator, head of institutional research, researcher, reviewer, auditor, and general visitor access rights —have been generated. All of these usecase have 100% of the expected output produced by the application.

Keywords : application development, litabmas management application, HMVC, CI

PENDAHULUAN

Pada era komputer dan internet saat ini, teknologi informasi telah masuk ke berbagai bagian kehidupan manusia, terutama di tingkat universitas, seperti dalam pelaksanaan pendidikan, penelitian dan pengabdian masyarakat. Kegiatan penelitian dan pengabdian masyarakat (litabmas) umumnya dikelola oleh LPPM (Lembaga Penelitian dan Pengabdian Masyarakat). Kegiatan ini bertujuan untuk mewedahi dosen agar dapat memberikan kontribusi terkait kepakaran mereka dan menggunakan ipteks untuk menyelesaikan masalah yang dihadapi masyarakat. Namun, pelaksanaan kegiatan Litabmas seringkali menghadapi permasalahan, terutama dalam hal



pengelolaan kegiatan, mulai dari pengumpulan proposal hingga pelaporan kegiatan. Apalagi jika pengelolaan pelaksanaan kegiatan litabmas ini masih memakai metode tradisional seperti yang dilakukan di LPPM UPN Veteran Jatim di beberapa periode ini. Proses penerimaan proposal hingga laporan akhir dan luaran, belum terdokumentasi dengan baik dan peneliti masih menyerahkan arsip proposal dan laporan berupa dokumen cetak. Permasalahan yang muncul adalah keterbatasan ruang penyimpanan arsip, arsip sering hilang atau rentan rusak. Permasalahan lainnya adalah LPPM mengalami kesulitan dalam hal pelacakan penelitian yang masih memiliki tanggungan luaran dari periode sebelumnya dan ketika ada pihak auditor yang memerlukan data penelitian, LPPM tidak dapat memberikan data dengan cepat.

UCB menghadapi permasalahan yang sejenis, sehingga untuk mengatasinya Manu & Tantrisna (2020) membuat aplikasi SIMLITABMAS untuk UCB dengan dua level pengguna: Administrator (LP3M) dan Dosen. Dengan aplikasi ini, dosen dapat mengelola penelitian dan pengabdianya sendiri, sedangkan LP3M memiliki akses untuk mengelola penelitian dan pengabdian secara keseluruhan. Aplikasi ini dibangun dengan mempergunakan PHPMaker 2020 dan *database* MySQL. Homaidi (2021) juga membuat aplikasi pengusulan dan sistem untuk memantau pelaksanaan penelitian pengabdian masyarakat di Universitas Ibrahimy sebagai solusi untuk permasalahan tersebut. Aplikasi yang dibuat berbentuk web dan dapat mengelola proses pengusulan proposal, penjadwalan seminar, dan pencatatan progres pelaksanaan. Pencatatan progres pelaksanaan meliputi catatan harian, laporan kemajuan, laporan akhir, dan laporan luaran hasil. Aplikasi ini dapat berfungsi sebagai gudang digital untuk data penelitian dan pengabdian masyarakat beserta luaran-luaran yang dihasilkan. Dengan demikian, pelaporan penelitian dan pengabdian masyarakat di Universitas Ibrahimy telah terdokumentasi dengan baik dan prosesnya lebih efisien dan efektif. Selain itu, data dapat dengan mudah diakses melalui website yang disediakan jika dibutuhkan.

Saat ini, banyak *framework* digunakan untuk mengembangkan aplikasi web, dan hampir semua *framework* web mengklaim mendukung pola desain MVC (Ma et al., 2019). Seringkali, dalam kerangka aplikasi web bergaya MVC, *model*, *controller*, dan *view* diatur secara terpisah. Namun, *controller* (dan *view*) untuk fitur aplikasi yang berbeda diatur dalam *namespace* yang sama. Ketika aplikasi menjadi lebih kompleks dan pembuatan aplikasi dilakukan secara berkolaborasi dalam bentuk tim pengembang, banyak masalah yang muncul. Untuk mengatasi masalah ini, berbagai varian dari MVC diusulkan. salah satunya adalah HMVC (Ma et al., 2019).

Salah satu pengembangan dari arsitektur *Model View Controller* (MVC) adalah *Hierarchical Model View Controller* (HMVC) yang terdiri dari beberapa MVC yang disusun menjadi satu kesatuan untuk aplikasi tertentu. Pola HMVC memungkinkan *programmer* membagi sistem ke dalam bentuk modul-modul yang lebih spesifik (Nurfauziah & Arjo, 2021), sehingga pengerjaan aplikasi menjadi lebih fleksibel dan lebih cepat (Hendra et al., 2022) karena memungkinkan beberapa *programmer* mengerjakan modul bagiannya masing-masing secara bersamaan. HMVC dapat diterapkan untuk membangun sistem yang lebih terstruktur dan mudah dikembangkan saat mengembangkan aplikasi manajemen Litabmas pendanaan mandiri. Dari sekian banyak *framework* PHP yang ada, CodeIgniter merupakan salah satu *framework* yang mendukung implementasi MVC (Bemindra & Nurhasanah, 2023; Purbo, 2021) dan juga mendukung implementasi HMVC (Hendra et al., 2022; Nurfauziah & Arjo, 2021). *Framework* ini juga memiliki dokumentasi yang cukup lengkap (Henri et al., 2020). Menimbang keunggulan tersebut, *framework* PHP yang dipilih untuk penelitian ini adalah CodeIgniter.

Aplikasi manajemen Litabmas pendanaan mandiri memiliki kebutuhan fungsional yang jelas dan dapat diidentifikasi di awal penelitian, sehingga pengembangan aplikasi akan dilakukan dengan metode *waterfall* (Henri et al., 2020; Thesing et al., 2021). Penelitian ini tidak menerapkan semua tahapan *waterfall* dan hanya berfokus pada tahapan permodelan dan pengembangan. Dengan adanya aplikasi ini, diharapkan proses pengajuan proposal Litabmas, pemantauan progres, dan pelaporan hasil dapat dilakukan secara sistematis dan jelas. Hal ini akan memberikan manfaat yang signifikan dalam meningkatkan akuntabilitas dan efektivitas kegiatan Litabmas secara keseluruhan.

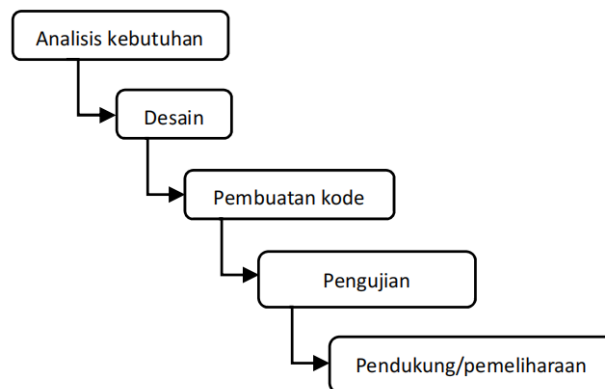
1. Hierarchical Model View Controller (HMVC)

Merupakan sebuah pengembangan dari arsitektur Model View Controller (MVC), *Hierarchical Model View Controller* (HMVC) terdiri dari beberapa MVC yang disusun menjadi satu kesatuan aplikasi. Struktur *model - Model, View, dan Controller - triad*, juga dikenal sebagai segitiga MVC, dibuat berlapis. Setiap lapisan *triad* MVC memiliki *Controller*, yang bertanggung jawab untuk mengakses lapisan *triad* MVC lainnya. Dengan kata lain, akses *Controller* memungkinkan komunikasi antar lapisan *triad* MVC. Kumpulan lapisan *triad* MVC yang saling terhubung ini membentuk hierarki, yang disebut HMVC (Hendra et al., 2022). Dengan memakai arsitektur seperti ini, pengerjaan aplikasi dapat dibagi-bagi kedalam beberapa *programmer* dan dikerjakan secara serentak (Hendra et al., 2022; Henri et al., 2020) karena setiap *triad* dapat berfungsi secara mandiri, tidak mempengaruhi *triad* lain.

2. Metodologi Waterfall

Waterfall merupakan metode SDLC yang dapat menjadi satu opsi metode pengembangan aplikasi, jika kebutuhan aplikasi sudah terdefiniskan dengan jelas dan stabil (tidak ada perubahan di tengah jalan) (Henri et al.,

2020; Thesing et al., 2021). Tahapan metode pengembangan ini meliputi tahapan analisa kebutuhan, desain, pembuatan kode, pengujian dan pendukung/pemeliharaan (Hendri et al., 2021; Indah, 2022; Laoli & Kristiana, 2022) seperti yang ditampilkan pada Gambar 1.



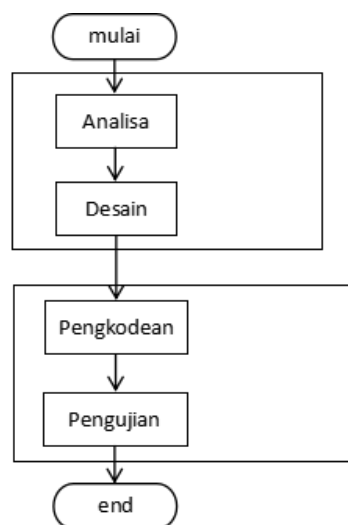
Sumber: Indah (2022)

Gambar 1. Metodologi Waterfall

Dalam tahapan komunikasi akan dilakukan inisiasi proyek dan pengumpulan kebutuhan awal. Selanjutnya adalah perencanaan, dimana akan dilakukan estimasi, penjadwalan dan *tracking* proyek. Dalam tahap pemodelan, akan dilakukan analisa dan desain terhadap aplikasi yang akan dibangun. Pengkodean dan pengujian aplikasi ada didalam tahapan pengembangan. Pengujian aplikasi ini dapat dilaksanakan dengan berbagai macam metode, baik secara *blackbox* (Hendra et al., 2022) maupun *whitebox*. Kedua metode ini dapat dilaksanakan secara manual (Fahrezi et al., 2022; Maulana et al., 2023; Supriyono, 2020) atau otomatis (Arfan & Hendrik, 2022; Asrin, 2023). Fokus penelitian ini adalah pada pengujian *blackbox* secara manual. Setelah aplikasi lolos uji, tahap selanjutnya adalah *delivery* aplikasi, pemberian dukungan teknis terkait operasional aplikasi (support dan feedback) dalam tahap *deployment*.

METODE PENELITIAN

Tahap-tahap dalam penelitian ini hanya menjalankan sebagian tahapan *waterfall* dari presmann, karena fokus dari penelitian ini adalah pengembangan aplikasi. Tahapan tersebut meliputi pemodelan dan pengembangan seperti yang ditampilkan pada Gambar 2.



Sumber : Penelitian (2023)

Gambar 2. Metode Penelitian

1. Analisa Kebutuhan

Untuk mendapatkan kebutuhan aplikasi, perlu diketahui permasalahan apa saja yang muncul dari pelaksanaan

litabmas. Permasalahan ini didapatkan dari wawancara dan pengamatan. Hasil wawancara dan pengamatan ini akan dikelompokkan dan dikategorikan, mana saja yang dapat diselesaikan dengan aplikasi. Dari proses ini, diperoleh kebutuhan fungsional dan nonfungsional aplikasi.

2. Desain

Setelah didapatkan kebutuhan fungsional dan non fungsional, langkah berikutnya adalah memodelkan dalam bentuk perancangan antarmuka, perancangan basis data dan juga perancangan proses.

3. Pengkodean

Tahapan berikutnya adalah mengimplementasikan rancangan dalam bentuk kode aplikasi. Pengkodean akan dilakukan dengan mempergunakan PHP *framework* CodeIgniter dengan memakai PostgreSQL sebagai *database* servernya. Konsep pengkodean mempergunakan HMVC

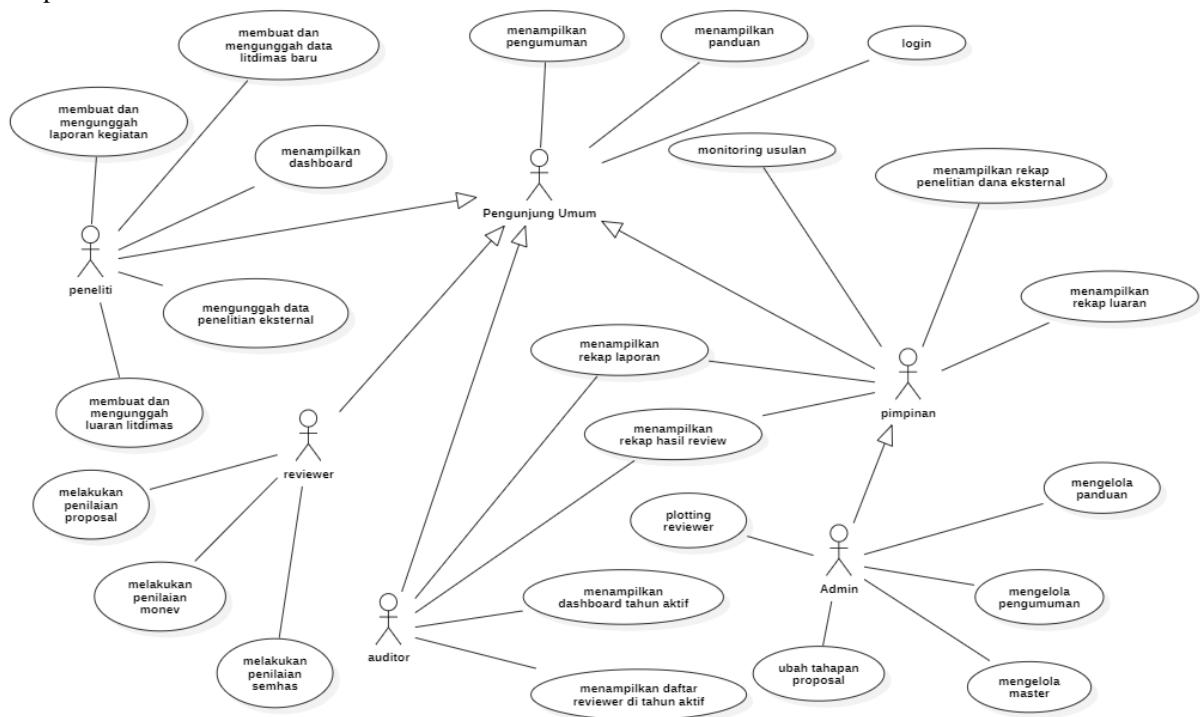
4. Pengujian

Setelah aplikasi selesai dikodekan, aplikasi akan diuji apakah sudah bebas kesalahan, apakah sudah memenuhi semua kebutuhan pengguna dll. Pengujian aplikasi akan dilakukan secara *blackbox*.

HASIL DAN PEMBAHASAN

1. Analisa Kebutuhan

Salah satu hasil dari tahap ini adalah kebutuhan fungsional yang digambarkan pada diagram *usecase*. Terdapat beberapa level pengguna (hak akses) dari aplikasi ini, yaitu pengunjung umum, peneliti, *reviewer*, admin, pimpinan dan juga level auditor. Detil fungsionalitas yang dimiliki untuk masing-masing hak akses ditunjukkan pada Gambar 3



Sumber : Penelitian (2023)

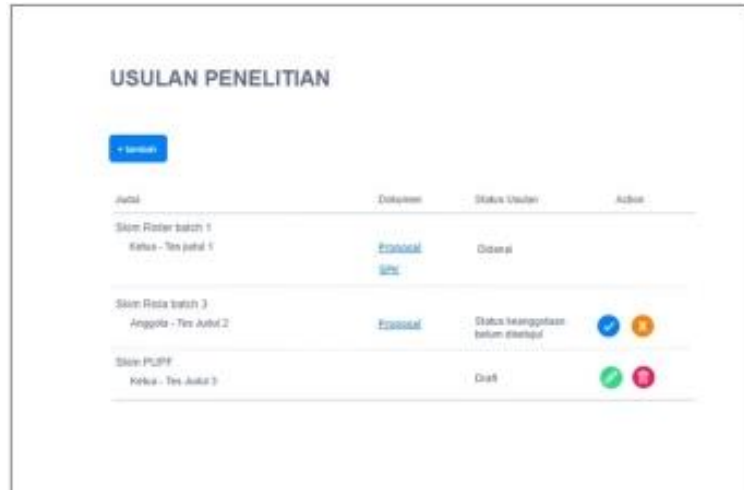
Gambar 3. Diagram *Usecase*

2. Desain

Tahap selanjutnya adalah memodelkan perancangan antarmuka dan perancangan proses untuk tiap *usecase* dan dilanjutkan dengan perancangan basis data. Uraian dari masing-masing proses sebagai berikut,

a. Perancangan antar muka

Untuk setiap *usecase*, akan dibuat rancangan antar muka tiap halamannya, gambar 4 dan 5 merupakan rancangan antar muka untuk *usecase* “membuat dan mengunggah data litdimas baru”. Proses ini dimulai dengan menampilkan daftar usulan penelitian yang dimiliki oleh seorang dosen, baik dosen tersebut berperan sebagai ketua maupun anggota suatu penelitian, seperti yang ditunjukkan pada gambar 4.



Sumber : Penelitian (2023)

Gambar 4. Rancangan Antar Muka untuk Menampilkan Data Usulan

Peneliti menekan tombol tambah, untuk membuat usulan baru, dan halaman berikutnya yang dimunculkan adalah halaman memasukkan informasi detail penelitian beserta *upload* file usulan seperti yang ditunjukkan pada gambar 5.



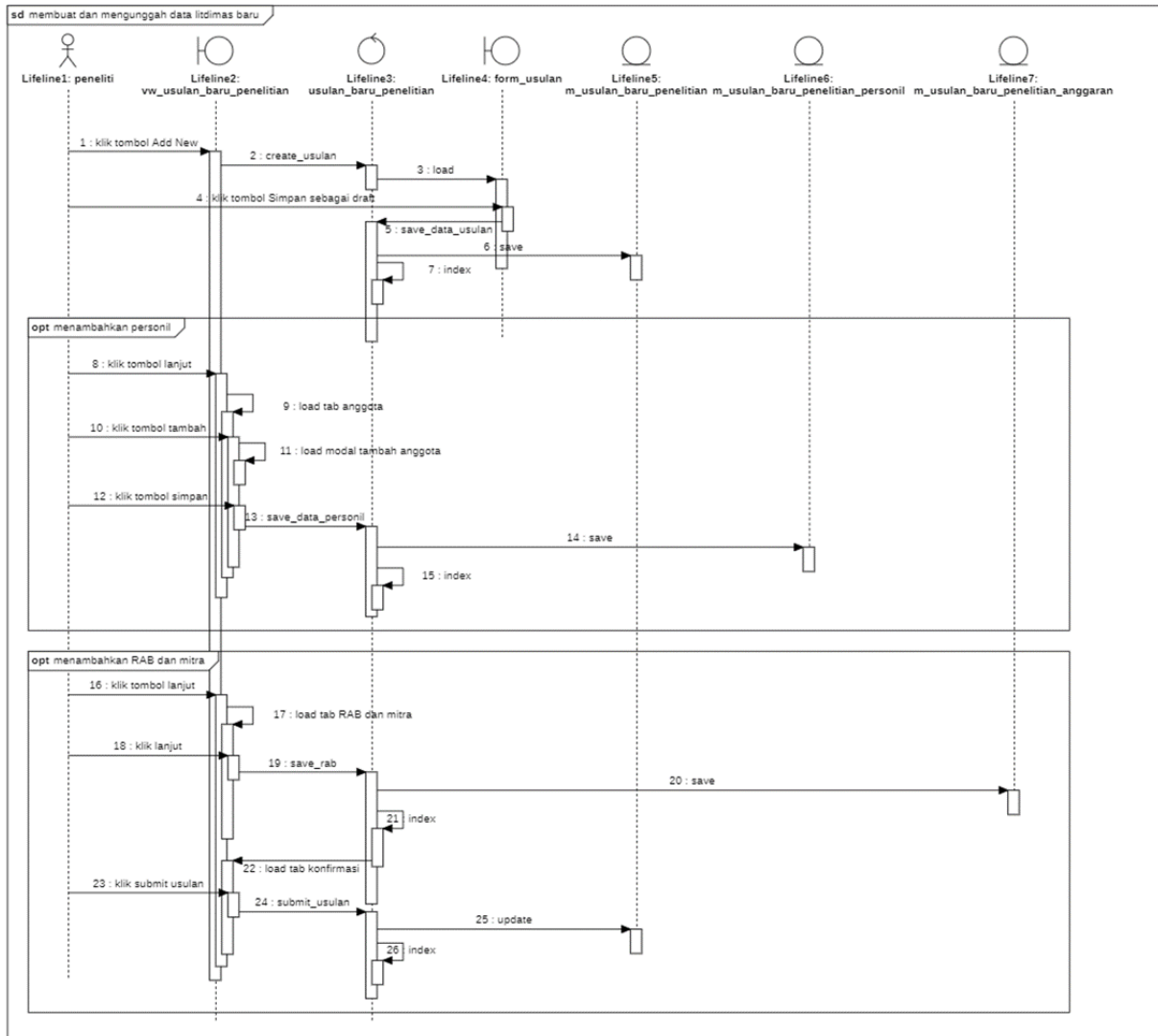
Sumber : Penelitian (2023)

Gambar 5. Rancangan Antar Muka untuk Menambahkan Data Usulan

Proses pembuatan rancangan antarmuka ini dilakukan untuk semua *usecase* yang ada, sebagai panduan bagi *programmer* untuk membangun aplikasi.

b. Perancangan proses

Proses berikutnya adalah membuat rancangan proses dalam bentuk diagram *sequence*. Diagram ini menggambarkan seluruh *class* yang dilibatkan untuk *usecase* “membuat dan mengunggah data litdimas baru”. Ada 6 *class* yang terlibat, dan interaksi keseluruhan *class*, beserta *method* yang dipanggil ditunjukkan pada gambar 6.



Sumber : Penelitian (2023)

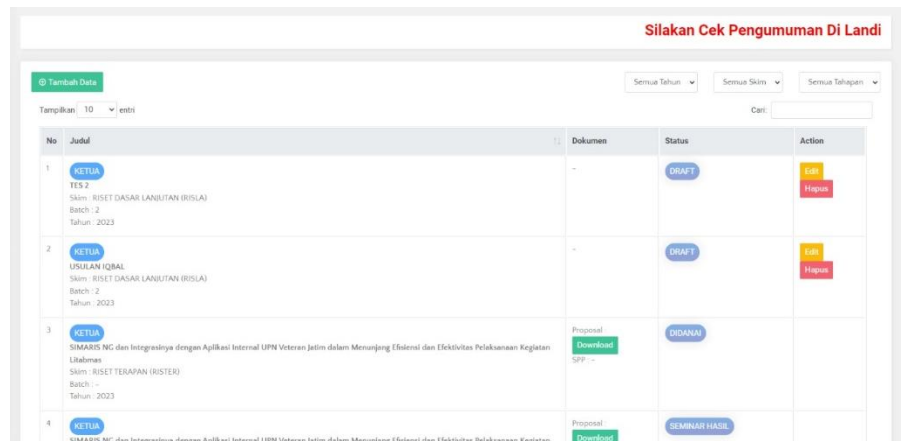
Gambar 6. Rancangan Antar Muka untuk membuat dan mengunggah data litdimas baru

c. Perancangan basis data

Perancangan basis data dilakukan dengan mempergunakan Power Designer. Terdapat 10 entitas utama, antara lain tabel skim, tahun_kegiatan, syarat_administrasi, syarat_anggaran, usulan, daftar_reviewer, plotting_reviewer, penilaian, jadwal_seminar, luaran.

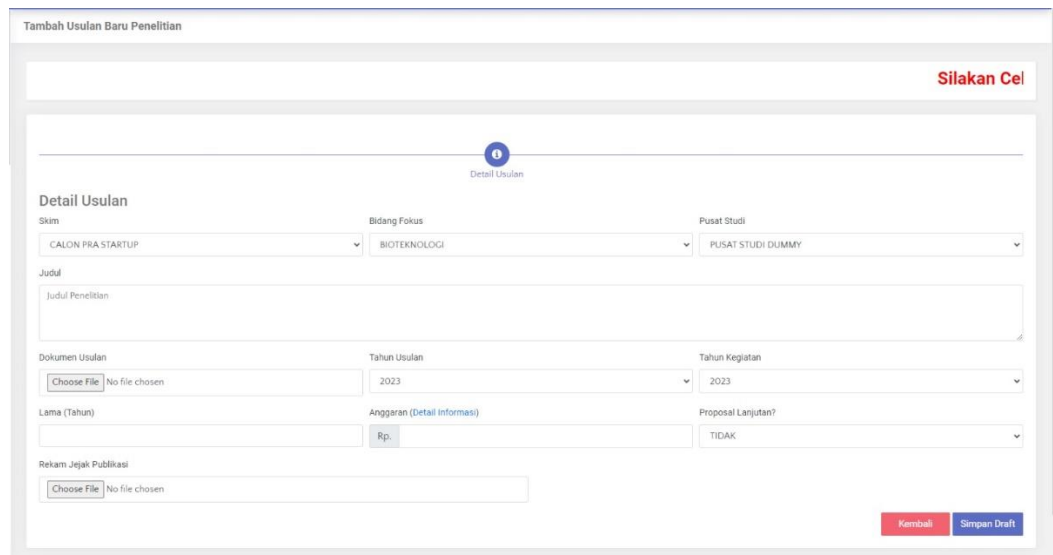
3. Pengkodean

Implementasi dari hasil pengkodean, untuk usecase “membuat dan mengunggah data litdimas baru” ditampilkan pada gambar 7 dan 8.



Sumber : Penelitian (2023)

Gambar 7. Hasil Impelementasi Kode untuk Menampilkan Data Usulan



Sumber : Penelitian (2023)

Gambar 8. Hasil Impelementasi Kode untuk Menambahkan Data Usulan

4. Pengujian

Dari 23 *usecase* diatas, masing masing dilakukan pengujian dengan metode *blackbox* mempergunakan berbagai macam skenario masukan. Hasil akhir dari tahap ini adalah 23 *usecase* yang sudah 100% memberikan *output* sesuai yang diharapkan. Tabel 1 berikut merupakan rangkuman hasil uji beberapa skenario untuk *usecase* menambahkan data usulan.

Tabel 1. Tabel Hasil Uji untuk usecase membuat dan mengunggah data litdimas baru

Sumber : Penelitian (2023)

No	Skenario penguian	Input	Hasil yang diharapkan	Hasil pengujian	kesimpulan
1	Isi tab detail usulan dengan semua inputan bernilai benar	Skim : rister Bidang Fokus : Bioteknologi Pusat studi : TIK Judul : tes judul rister dokumen usulan : file pdf kurang dari 5 MB	Berhasil menyimpan data ke <i>database</i>	Sesuai	Berhasil

		Tahun usulan : 2023 tahun kegiatan : 2023 Lama tahun : 1 anggaran : 24.000.000 Proposal lanjutan : tidak Rekam jejak publikasi : <i>upload</i> file pdf kurang dari 5 MB			
2	Isi tab detail usulan dengan beberapa inputan bernilai salah	Skim : rister Bidang Fokus : Bioteknologi Pusat studi : TIK Judul : tes judul rister1 dokumen usulan : file pdf lebih dari 5 MB Tahun usulan : 2023 tahun kegiatan : 2023 Lama tahun : 1 anggaran : 4.000.000 Proposal lanjutan : tidak Rekam jejak publikasi : <i>upload</i> file pdf lebih dari 5 MB	Tetap menampilkan halaman tab detail usulan, dengan pesan tambahan “file berupa pdf dan ukuran file harus dibawah 5 MB” serta pesan “anggaran yang dimasukkan tidak sesuai dengan skim yang dipilih”	Sesuai	Berhasil

KESIMPULAN

Aplikasi manajemen Litabmas pendanaan mandiri berhasil dikembangkan dengan memakai metode *waterfall*. Bahasa Pemrograman PHP dengan memakai *Framework* CodeIgniter dan database PostgreSQL dipergunakan pada saat pengkodean. Struktur penulisan kode pada saat pengkodean, dibangun dengan memakai konsep arsitektur HMVC untuk memudahkan pengerjaan secara kolaboratif. Sebelum dioperasionalkan, aplikasi diuji mempergunakan metode *blackbox*, dan menghasilkan 23 *usecase* dari 6 level pengguna - admin, pimpinan, peneliti, *reviewer*, *auditor* dan pengunjung umum - yang sudah 100% memberikan *output* sesuai dengan yang diharapkan. Aplikasi ini dapat mempermudah LPPM dan dosen dalam mengelola kegiatan litabmas yang berasal dari skim pendanaan mandiri, meliputi pembuatan usulan litabmas, *ploting reviewer*, penilaian proposal oleh *reviewer*, upload laporan kemajuan, laporan pendanaan, luaran dan penilaian hasil penelitian oleh *reviewer* (monev dan semhas). Aplikasi ini juga memberikan akses untuk pihak eksternal dalam hal ini *auditor*) dalam menampilkan rekap laporan, menampilkan rekap hasil *review*, menampilkan *dashboard* tahun aktif dan menampilkan daftar *reviewer* di tahun aktif.

REFERENSI

- Arfan, A., & Hendrik, H. (2022). Penerapan STLC dalam Pengujian Black Box dengan Automation Testing Tool (Studi kasus: PT.GIT Solution). *AUTOMATA*, 3(2). <https://journal.uui.ac.id/AUTOMATA/article/view/24127>
- Asrin, F. (2023). Black Box Testing of Futsal Field Rental Information Systems Using Automated Testing Method. *Journal of Information Systems and Informatics*, 5(3), 928–955. <https://doi.org/10.51519/journalisi.v5i3.540>
- Bemindra, J. A., & Nurhasanah. (2023). Penerapan Model View Controller Dengan Framework CodeIgniter Pada E-Commerce Penjualan F&B (Studi Kasus: Wardel). *LOGIC : Jurnal Ilmu Komputer Dan Pendidikan*, 1(3), 617–629. <https://journal.mediapublikasi.id/index.php/logic>
- Fahrezi, A., Salam, F. N., Ibrahim, G. M., Syaiful, R. R., & Saifudin, A. (2022). Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia. *LOGIC : Jurnal Ilmu Komputer Dan Pendidikan*, 1(1), 1–5. <https://journal.mediapublikasi.id/index.php/logic>
- Hendra, Y., Tarigan, R., & Usman, A. U. (2022). Pengembangan Sistem Administrasi TOEFL Menggunakan Arsitektur HMVC (Hierarchical Model-View-Controller) Dan Framework CodeIgniter. *JSiI (Jurnal Sistem Informasi)*, 9(2), 177–185. <https://doi.org/10.30656/jsii.v9i2.4790>

- Hendri, Jefi, Kholifah, D. N., Solecha, K., & Armanda, R. N. (2021). Perancangan Aplikasi Penjualan Furniture Dengan Metode Waterfall Berbasis Web. *Computer Science (CO-SCIENCE)*, 1(2), 161–170. <http://jurnal.bsi.ac.id/index.php/co-science>
- Henri, G. J., Rusdianto, D. S., & Jonemaro, E. M. A. (2020). Pengembangan Aplikasi Sistem Informasi Manajemen Laboratorium Teknik Sumberdaya Alam dan Mineral (TSAL) Fakultas Teknologi Pertanian Universitas Brawijaya. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(5), 1417–1425. <http://j-ptiik.ub.ac.id>
- Homaidi, A. (2021). Aplikasi Pengusulan dan Pemantauan Pelaksanaan Penelitian dan Pengabdian Masyarakat Universitas Ibrahimy. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 20(2), 225–236. <https://doi.org/10.30812/matrik.v20i2.942>
- Indah, N. N. (2022). Analisis Dan Perancangan Sistem Informasi Pengelolaan Arus Kas Pada PT Graha Sentramulya. *Ilmudata.Org*, 2(2), 1–13.
- Laoli, D., & Kristiana, T. (2022). Sistem Informasi Pemasaran Perumahan Pada PT. Trixie Graha Anugerah Berbasis Website Menggunakan Metode Waterfall. *Computer Science (CO-SCIENCE)*, 2(2), 143–152. <http://jurnal.bsi.ac.id/index.php/co-science>
- Ma, M., Yang, J., Wang, P., Liu, W., & Zhang, J. (2019). Light-Weight and Scalable Hierarchical-MVC Architecture for Cloud Web Applications. *Proceedings - 6th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2019 and 5th IEEE International Conference on Edge Computing and Scalable Cloud, EdgeCom 2019*, 40–45. <https://doi.org/10.1109/CSCloud/EdgeCom.2019.00017>
- Manu, G. A., & Tantrisna, E. (2020). Perancangan Aplikasi Monitoring Penelitian Dan Pengabdian Masyarakat Internal Perguruan Tinggi. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, 3(2), 48–55. <https://phpmaker.dev/download.php>
- Maulana, B. A., Mawarni, E., Hidayattuloh, M. Y., Suryawijaya, V., & Saifudin, A. (2023). Pengujian Black Box pada Sistem Informasi Barang Berbasis Web Menggunakan Metode Boundary Value Analysis. *OKTAL : Jurnal Ilmu Komputer Dan Science*, 2(6), 17477–1753.
- Nurfauziah, S., & Arjo, T. R. (2021). Pengembangan Sistem Informasi Pendaftaran Praktek Kerja Lapangan (PKL) dengan Konsep Hierarchical Model View Controller (HMVC) Studi Kasus: Jurusan Administrasi Niaga Politeknik Negeri Malang. *Journal of Applied Informatics and Computing (JAIC)*, 5(1), 28–36. <http://jurnal.polibatam.ac.id/index.php/JAIC>
- Purbo, O. W. (2021). A Systematic Analysis: Website Development using Codeigniter and Laravel Framework. *Enrichment: Journal of Management*, 12(1), 1008–1014.
- Supriyono. (2020). Software Testing with the approach of Blackbox Testing on the Academic Information System. *International Journal of Information System & Technology*, 3(2), 227–233.
- Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus Waterfall Project Management: Decision model for selecting the appropriate approach to a project. *Procedia Computer Science*, 181, 746–756. <https://doi.org/10.1016/j.procs.2021.01.227>