

Implementasi *Model-View-Template* Pada Aplikasi Surat Keterangan Pendamping Ijazah Berbasis *Web*

Yeyen Dwi Atma^{1*}, Pramudya Prima Insan², Nasruddin Bin Idris³

^{1*,2,3}Program Studi Teknologi Informasi, Universitas Mulia, Indonesia

*Email: yeyen@universitasmulia.ac.id

Abstract

Based on KKNI capabilities, each graduate is entitled to induce an SKPI agreeing to their uncommon abilities. The reason of this research is to plan the SKPI application at Mulia University. This SKPI application advancement strategy employs the model strategy with UML modeling and the use of the django system with the model-view-template concept. With this SKPI framework, it is trusted that all imminent graduates of Mulia University] who will graduate can fill in and transfer all accomplishments and certificates agreeing to their field of information. The designer program overseeing imminent graduates can also confirm SKPI input online. With this web-based SKPI framework, it is trusted that it can make it less demanding for graduates to get SKPI and the authoritative administration of SKI issuance runs well.

Keywords: SKPI, web, django, mvt, prototype.

Abstrak

Berdasarkan kualifikasi KKNI setiap lulusan berhak mendapatkan SKPI sesuai keahlian khusus yang dimilikinya. Tujuan penelitian ini adalah merancang bangun aplikasi SKPI pada Universitas Mulia. Metode pengembangan aplikasi SKPI ini menerapkan metode purwarupa dengan pemodelan UML dan penggunaan framework django dengan konsep model-view-template. Dengan sistem SKPI ini diharapkan seluruh calon lulusan Universitas Mulia yang akan yudisium dapat mengisi dan mengunggah semua prestasi dan sertifikat sesuai bidang ilmunya, program studi pengelola calon lulusan juga dapat memverifikasi inputan SKPI secara online. Dengan adanya sistem SKPI berbasis web ini diharapkan dapat mempermudah lulusan memperoleh SKPI dan pengelola administrasi penerbitan SKPI berjalan lancar.

Kata kunci: SKPI, framework, django, mvt, prototype.

1. Pendahuluan

Dalam amanat Kurikulum yang termuat pada Kerangka Kualifikasi Nasional Indonesia (KKNI), setiap lulusan atau calon sarjana baru berhak mendapatkan SKPI [1]. Hal ini dilakukan agar selain lulusan mendapatkan ijazah formal, lulusan juga memiliki SKPI yang memuat keahlian yang dimiliki lulusan.

Berlandaskan Peraturan Menteri Pendidikan dan Kebudayaan No.73 Tahun 2013 [2], Peraturan Mendikbud No. 49 Tahun 2014, dan Peraturan Mendikbud No. 81 Tahun 2014 [3]. Dengan terbitnya SKPI, maka peraturan KKNI tersebut secara substansial membawa peningkatan sistem penjaminan mutu yang bisa menjalankan fungsi memantau

dan peninjauan terhadap Perguruan tinggi penghasil lulusan.

Aturan yang diterbitkan mengharuskan pihak Perguruan Tinggi menelaah dan menerbitkan SKPI berdasarkan capaian pembelajaran, prestasi dan keahlian mahasiswa selama menempuh perkuliahan [4]. Dalam hal penerbitan SKPI ini tentunya pengelola akan sangat kesulitan jika harus menelaah satu persatu secara manual berdasarkan berkas fisik.

Berlandaskan hal tersebut, untuk melaksanakan proses penerbitan SKPI dibutuhkan sistem operasi standar dan sistem informasi online yang dapat membantu dan mempermudah proses administrasi staff Perguruan Tinggi dan calon lulusan.

Melalui sistem SKPI berbasis web, seluruh mahasiswa calon lulusan yang telah mengerjakan dan merampungkan tugas akhir dapat langsung menyelesaikan pengisian SKPI sesuai isian yang diminta, diantaranya data pelatihan, prestasi, skripsi, dan magang. Selanjutnya program studi verifikasi isian SKPI secara online.

2. Metoda Penelitian

2.1. Metode Pengumpulan Data

Dalam penelitian ini, menerapkan tiga teknik digunakan untuk mengumpulkan data yaitu.

a. Observasi

Dalam penelitian ini, penulis melakukan observasi dengan cara mengamati secara langsung proses pengelolaan dan pengadministrasian dokumen SKPI di Universitas Mulia.

b. Wawancara

Penulis melakukan wawancara dengan pihak akademik dan program studi di lingkungan Universitas Mulia. Tujuan dari wawancara ini adalah untuk mengkonfirmasi data atau fakta yang telah diperoleh selama pengamatan lapangan terkait proses bisnis pengelolaan dan pengadministrasian SKPI. Dengan melakukan wawancara, penulis dapat memperoleh informasi langsung dari

pengguna dan pelaku proses bisnis tersebut.

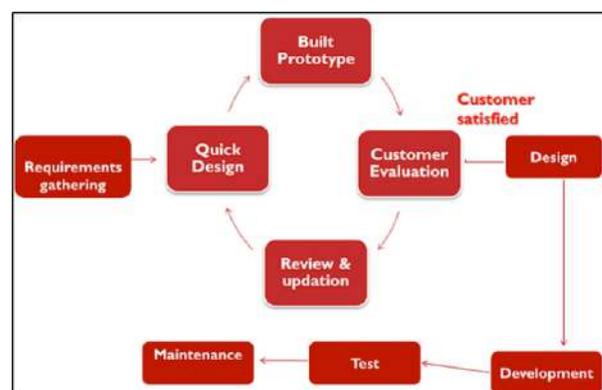
c. Studi Pustaka

Penulis melakukan studi pustaka dengan membaca buku-buku literatur django, MVT, dan laporan penelitian sesuai kebutuhan.

2.2. Metode Pengembangan Sistem

Dalam penelitian ini, pengembangan aplikasi menggunakan kerangka pengembangan perangkat lunak prototype. Prototipe merupakan versi awal dari perangkat lunak yang digunakan untuk mewakili percobaan desain yang bertujuan untuk memudahkan pengembang dan pengguna dalam menemukan solusi perancangan sistem yang sesuai dengan kebutuhan.

Terdapat 6 tahapan model prototipe dalam *software development lifecycle* seperti ditunjukkan pada Gambar 1. Langkah pertama dilakukan pengumpulan dan menganalisa persyaratan pengguna, kedua membuat desain cepat atau desain kasar, ketiga membangun prototipe, keempat melakukan evaluasi kebutuhan atau penilaian pengguna, langkah kelima melakukan penyempurnaan prototipe, langkah keenam melakukan penerapan sistem dan *maintenance* produk.



Gambar 1. Metode Pengembangan Prototipe

2.3. Model-View-Template

Django framework dikembangkan dengan mengadopsi arsitektur *Model View Template* (MVT), yang merupakan pola desain perangkat lunak yang digunakan dalam pengembangan

aplikasi berbasis web. Struktur MVT terdiri dari komponen utama yang saling terhubung sebagai berikut [5]:

a. Model

Model berfungsi untuk mengelola data aplikasi dan menyediakan interaksi dengan database. Model ini menyimpan informasi mengenai objek, hubungan antara objek, dan aturan validasi data.

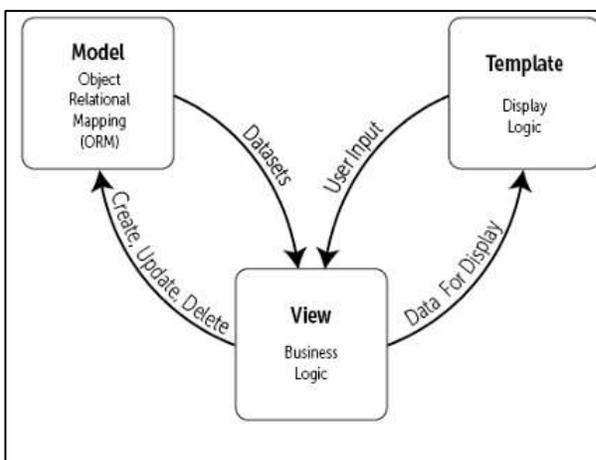
b. View

View bertanggung jawab untuk mengatur logika bisnis dan menangani permintaan dari pengguna. View ini mengambil data dari model dan memformatnya ke dalam bentuk yang sesuai untuk ditampilkan kepada pengguna.

c. Template

Template bertugas untuk menghasilkan tampilan yang diberikan kepada pengguna. Template menggunakan bahasa template yang terintegrasi dengan django untuk menggabungkan data dari model dengan markup HTML, sehingga menghasilkan tampilan yang dinamis.

Dengan adanya struktur MVT, jango *framework* menyediakan kerangka kerja yang kuat dan terstruktur untuk mengembangkan aplikasi web yang efisien dan mudah dipelihara. Berikut gambar arsitektur MVT pada django *framework* [6].



Gambar 2. Arsitektur Model-View-Template

Pada Gambar 2 untuk arsitektur model MVT menjelaskan bahwa masing-masing komponen memiliki fungsi dan fitur yang berbeda-beda, dengan memusatkan fokus pembentukannya dalam pengkodean sebelum implementasi dapat dikelola lebih mudah dan juga efisiensi untuk maintenance lebih cepat [6].

3. Hasil dan Pembahasan

3.1. Kebutuhan Sistem

Analisis kebutuhan sistem perlu dilakukan guna mengetahui sasaran atau target dari tujuan akhir sebuah aplikasi [7]. Dalam rancangan aplikasi SKPI, didapatkan melalui penggalian informasi mengenai standar operasional prosedur (SOP) dan mekanisme di lapangan yang akan diterapkan menjadi produk baru dalam sistem SKPI berbasis web.

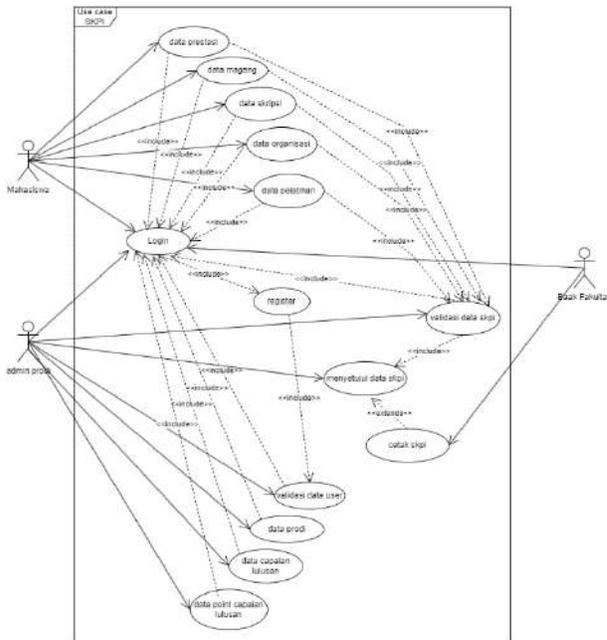
Tabel 1. Kebutuhan Fungsional

No.	Kebutuhan Fungsional
1	Bagian Akademik dapat login kedalam sistem untuk menyimpan, menampilkan, mengubah data dan instansi secara lengkap dan user sesuai tingkatan level usernya.
2	Admin staf dapat melihat dan merubah data personal. Data capaian pembelajaran dan data lain-lain yang berhubungan dengan SKPI
3	Bagian akademik dapat mengelola dan melakukan monitoring jumlah data mahasiswa lulusan dan dokumen SKPI yang diselesaikan oleh mahasiswa
4	Sistem dapat memberikan informasi detail mengenai data personal mahasiswa, data prestasi, sertifikat dan dokumen SKPI.
4	Mahasiswa dapat mengisi dan mengirimkan data personal, prestasi, pelatihan, magang, dan validasi isian sebelum diperiksa oleh admin

3.2. Desain Pemodelan

Perancangan aplikasi SKPI ini mengikuti pemodelan berorientasi objek yaitu merancang fungsionalitas sistem untuk mengetahui tolak ukur sebuah capaian yang ingin dikembangkan dalam sistem untuk pengguna

[8]. Pemodelan sistem menggunakan teknik UML dapat membuat proses perancangan lebih mudah dipahami [9]. Dengan adanya representasi dalam bentuk Gambar ataupun diagram. dengan salah satunya pembentukan use-case diagram untuk deskripsi pengguna dalam aplikasi SKPI.



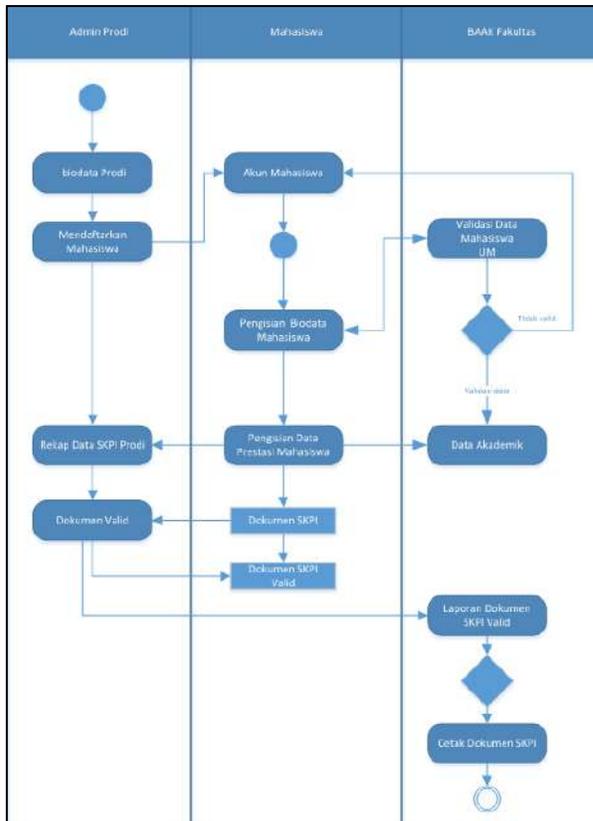
Gambar 3. Use Case Diagram SKPI

Seperti yang terlihat pada Gambar 3 diatas, use case diagram memiliki tiga user yakni bagian administrasi akademik fakultas (BAAK), admin program studi dan mahasiswa sebagai pemeran utama dari aplikasi untuk menjalankan proses bisnis sistem agar tujuan akhir dari aplikasi tercapai yaitu cetak dokumentasi digital untuk SKPI Mahasiswa.

Tabel 2. Uraian Use Case Pengguna

Use Case	Aktor	Keterangan
Registrasi User	Mahasiswa	Mahasiswa melakukan pendaftaran user aplikasi SKPI pada admin Prodi
Approval User	Admin	Admin melakukan approval pendaftaran user
Input data diri	Mahasiswa	Mahasiswa melakukan inputan data profil mahasiswa
Input data pelatihan	Mahasiswa	Mahasiswa melakukan inputan/upload data profil pelatihan
Input data organisasi	Mahasiswa	Mahasiswa melakukan inputan/upload data keikutsertaan dalam organisasi
Input data skripsi	Mahasiswa	Mahasiswa melakukan inputan data skripsi yang sudah diselesaikan
Input data magang	Mahasiswa	Mahasiswa melakukan inputan data magang yang sudah diselesaikan
Approval pengajuan SKPI	Admin	Admin verifikasi dan menyetujui pengajuan SKPI oleh lulusan
Pencetakan SKPI	Admin	Admin melakukan pencetakan setelah data valid

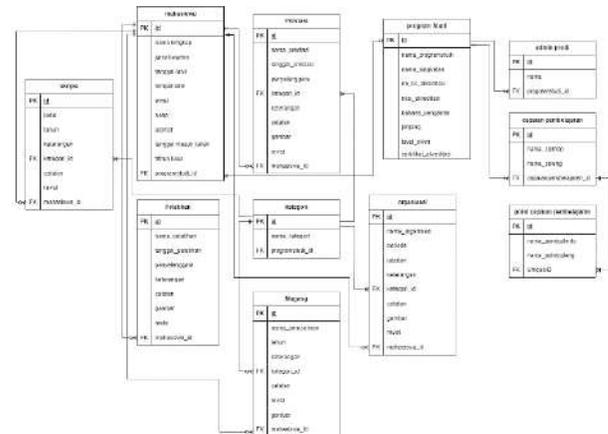
Kemudian pada perancangan activity diagram, mahasiswa melakukan pencatatan data pelatihan, prestasi, keanggotaan organisasi, skripsi, dan magang. Menunjukkan bukti dengan mengunggah dokumen pendukung, kemudian pengelola menindaklanjuti terhadap unggahan data tersebut. Gambar 4 menunjukkan hasil rancangan activity diagram aplikasi.



Gambar 4. Activity Diagram SKPI

Pada Gambar 4 terlihat dari proses-proses bisnis sistem yang dilakukan oleh user. Aktivitas dari user tersebut melakukan interaksi terhadap aplikasi untuk menjalankan fungsional aplikasi [10]. Secara prosedur, dokumen SKPI harus melalui proses validasi oleh pihak prodi hingga fakultas agar tetap melakukan pengecekan data yang telah dilakukan mahasiswa input [11].

Pada class diagram tersebut menggambarkan keterhubungan dengan bagian-bagian dari sistem SKPI. Relasi antar class mahasiswa memiliki relasi dengan class pelatihan, organisasi, magang dan skripsi. Class program studi memiliki relasi dengan class capaian pembelajaran, point capaian pembelajaran, dan konfirmasi pengajuan. Class Admin memiliki relasi dengan class pengajuan.



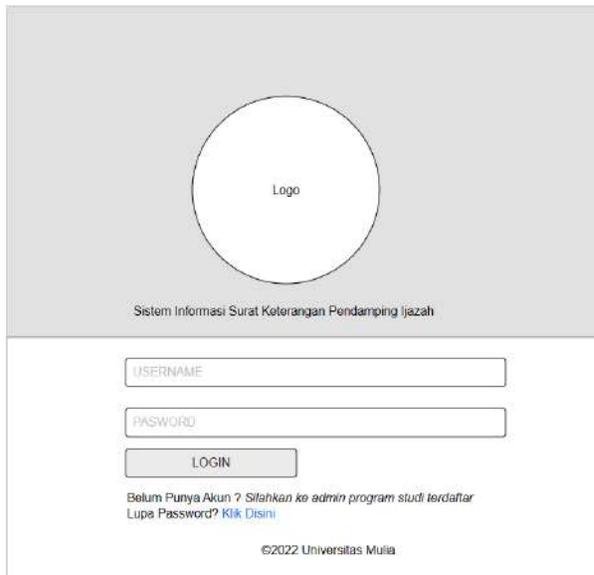
Gambar 5. Class Diagram SKPI

Pada Gambar 5, terlihat dalam class diagram untuk aplikasi SKPI telah terbentuk 8 entitas dengan korelasi database dengan 3 data tabel yang menjadi bagian pendukung untuk proses implementasi menjadi sebuah aplikasi [12].

3.3. Desain Purwarupa

Desain purwarupa atau yang lebih dikenal dengan *low-fidelity* dibuat untuk menghasilkan gambaran pemetaan tata letak fitur dalam aplikasi sehingga lebih memudahkan dalam mengimplementasikan sistem untuk SKPI [13]. Desain purwarupa aplikasi dibuat menggunakan *tools* yang secara khusus diperuntukan membuat desain *prototype* sebuah aplikasi. Pada rancangan desain ini digunakan sebuah aplikasi online yaitu *app.moqups.com*, dengan kelebihan fitur *stencil* yang dimiliki dapat melakukan perubahan secara realtime dan autosave dalam aplikasi berbasis website tersebut [14].

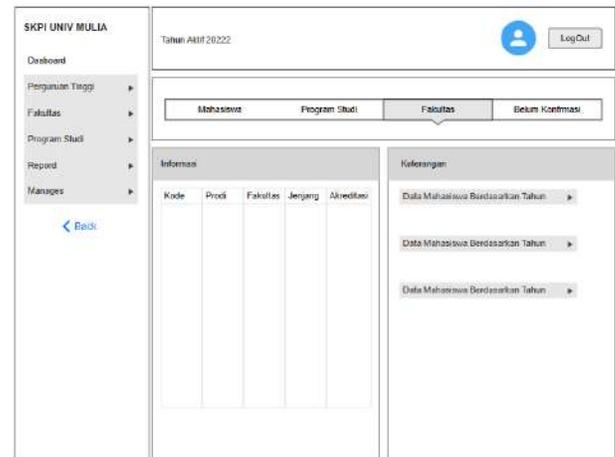
Hasil desain purwarupa dengan mengintegrasikan dua buah halaman utama, yaitu halaman admin dan halaman mahasiswa, pada masing-masing halaman memiliki perbedaan menu serta fungsi yang telah disesuaikan dengan kebutuhan pengguna secara fungsional sistem pada halaman dashboard.



Gambar 6. Desain Login SKPI

Halaman admin dirancang untuk staf administrasi atau petugas yang bertanggung jawab atas pengelolaan dan pemrosesan data SKPI [15]. Pada halaman ini, admin memiliki akses penuh untuk mengelola data mahasiswa, termasuk informasi pribadi, riwayat akademik, dan pencapaian non-akademik. Admin juga dapat mengelola *template* SKPI, menambahkan atau menghapus inputan yang diperlukan, serta mengatur proses verifikasi dan validasi data yang telah mahasiswa input ke dalam aplikasi SKPI [16].

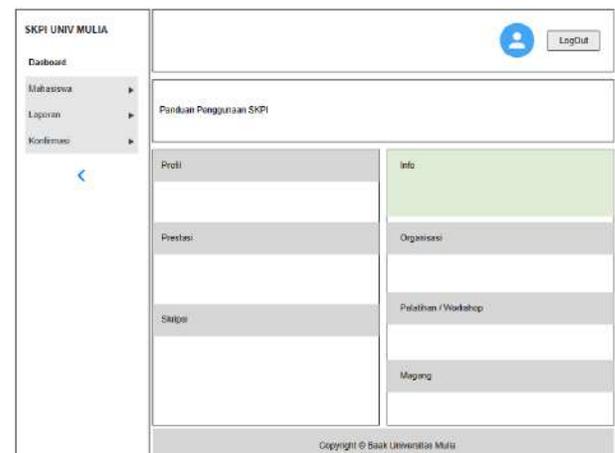
Selain itu, halaman admin juga harus dilengkapi dengan adanya fitur monitoring [17]. Sehingga memungkinkan admin untuk melacak status pengisian SKPI oleh mahasiswa, memberikan masukan atau petunjuk jika diperlukan, dan melakukan verifikasi terhadap data yang telah diinput oleh mahasiswa. Adapun hasil rancangan purwarupa untuk dashboard admin ditunjukkan pada Gambar 7.



Gambar 7. Desain Dashboard Admin SKPI

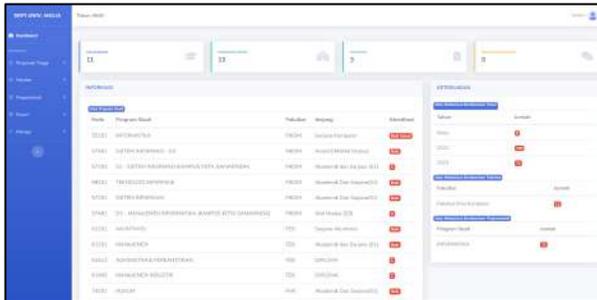
Selanjutnya halaman untuk *dashboard* mahasiswa, merupakan antarmuka yang ditujukan untuk mahasiswa yang sedang atau telah menyelesaikan studi di Perguruan Tinggi. Pada halaman ini, mahasiswa dapat mengakses berbagai menu yang memungkinkan mereka untuk mengisi dan memperbarui data SKPI mereka sendiri, termasuk pencapaian akademik, partisipasi dalam kegiatan ekstrakurikuler, magang, dan proyek riset.

Selain itu, halaman mahasiswa juga dilengkapi dengan fitur monitoring yang memungkinkan mahasiswa untuk melihat progres pengisian SKPI mereka, memperoleh *feedback* dari staf administrasi, serta memonitor status verifikasi dan validasi data oleh pihak berwenang Perguruan Tinggi [18].



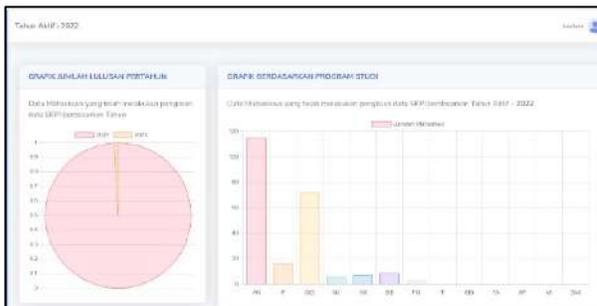
Gambar 8. Desain *Dashboard* Mahasiswa

perlu di *maintenance* atau data isian SKPI mahasiswa yang belum dikonfirmasi admin program studi. Selain itu pada halaman admin ini juga disediakan beberapa menu *report* yang bisa digunakan untuk melihat progres dan view SKPI dapat dilihat pada Gambar 14.



Gambar 14. Halaman Dashboard Admin SKPI

Pada tampilan halaman *dashboard* (Gambar 14), terlihat beberapa data dari hasil inputan *user* pengguna lainnya yang telah berhasil di input. Data yang terlihat pada halaman *dashboard* merupakan data dari masing-masing program studi yang dibentuk adalah untuk memudahkan pihak admin dalam monitoring serta adanya sistem kontrol secara internal universitas [20]. Selain itu terdapat proses monitoring tambahan dari aplikasi untuk user admin. Dimana pada Gambar 15 terlihat tampilan lain dari halaman dashboard admin SKPI untuk melihat data secara diagram.

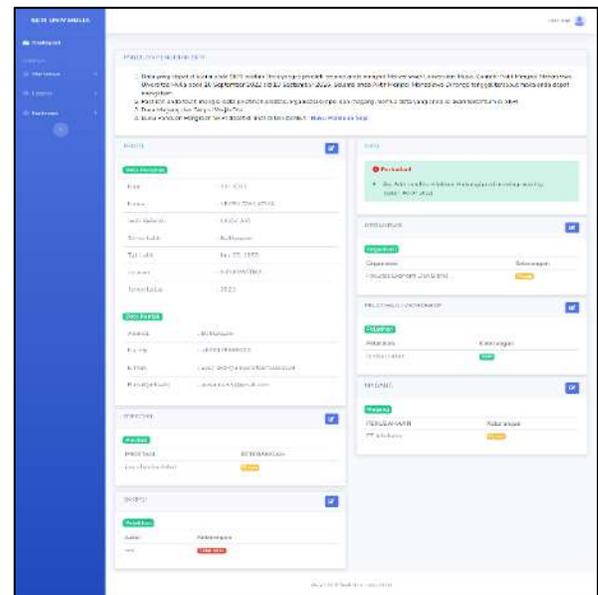


Gambar 15. Monitoring List Data SKPI Mahasiswa

Proses monitoring seperti pada Gambar 15 merupakan hasil pembentukan implementasi *view* dari data-data yang telah diinputkan oleh user lainnya. Representasi grafik tersebut merupakan grafik yang terbentuk berdasarkan data program studi pada Gambar 14

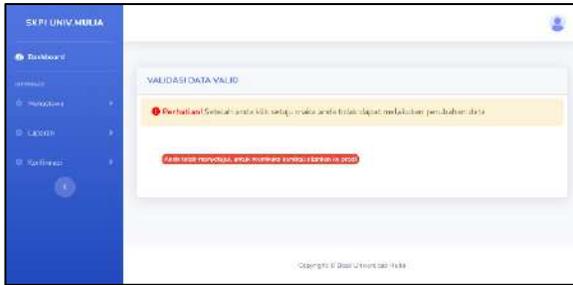
sebelumnya. Kebutuhan proses pembentukan visual data dari *requesting* database yang telah diintegrasikan antar *user*. Pentingnya monitoring dengan representasi visualisasi data adalah untuk memenuhi kriteria kelayakan sebuah aplikasi yang *suitability* [21].

Halaman mahasiswa, setelah mahasiswa berhasil login akan langsung mengarah ke halaman *dashboard* mahasiswa yang didalamnya terdapat menu yang digunakan untuk pengisian tentang informasi identitas diri, isian prestasi keahlian dan organisasi, dan juga upload berkas pendukungnya dapat dilihat pada Gambar 16.

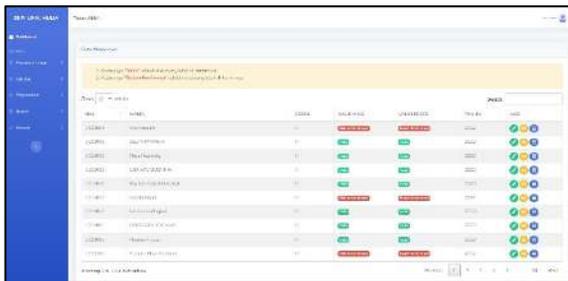


Gambar 16. Hasil Cetak Dokumen SKPI

Halaman konfirmasi SKPI. Setelah melakukan pengisian data SKPI mahasiswa. Mahasiswa diwajibkan konfirmasi atas pengisian yang sudah dilakukan, setelah mahasiswa konfirmasi data yang sudah diisi akan masuk ke halaman admin untuk dilakukan validasi keabsahan data pada Gambar 17 dan 18.



Gambar 17. Konfirmasi dan Validasi Data SKPI Mahasiswa



Gambar 18. Monitoring List Data SKPI Mahasiswa

Setelah itu proses selanjutnya adalah cetak dokumen SKPI. Pada halaman *view* SKPI baik mahasiswa, admin, dan fakultas bisa melihat isian data mahasiswa yang telah *valid*, kemudian dilakukan cetak dokumen sesuai format SKPI. Hasil cetak dokumen SKPI ditunjukkan pada Gambar 19.



Gambar 19. Hasil Cetak Dokumen SKPI

Hasil cetak dokumen SKPI pada Gambar 19 merupakan hasil pembentukan *view* yang telah dibuat. Format yang digunakan sesuai dengan standar nasional [4], maka tampilan pada hasil output pada halaman tersebut dapat secara langsung dicetak oleh masing-masing pengelola administrasi program studi sebagai dokumentasi.

Dokumen SKPI yang telah dicetak oleh aplikasi seperti pada Gambar 19 merupakan salah satu contoh luaran sistem. Selanjutnya yang harus dilakukan adalah penyebaran seluruh *user* terhadap *user* sebagai pengelola program studi dan juga mahasiswa setelah dibuat oleh masing-masing admin program studi di Universitas Mulia.

4. Kesimpulan

Struktur MVT dalam *django framework* memisahkan tugas-tugas yang berbeda secara jelas, sehingga secara keseluruhan, penggunaan *Model, View, dan Template* dalam *django framework* membantu pengembang dalam merancang dan mengembangkan aplikasi berbasis web yang terstruktur, efisien, dan mudah dipelihara. Selanjutnya diharapkan aplikasi SKPI Universitas Mulia ini dapat memenuhi kebutuhan proses bisnis penerbitan SKPI bagi mahasiswa calon lulusan dan mempermudah semua pihak dalam pengelolaan dokumen administrasi, monitoring, serta penerbitan SKPI.

5. Saran

Diperlukan integrasi data dari sistem utama sehingga data yang dihasilkan akurat dan cepat karena mahasiswa tidak perlu rekam ulang data-data mahasiswa yang sudah ada di sistem utama. Selanjutnya terkait isu faktor skalabilitas yang semakin lama semakin banyak pengguna yang melakukan pengisian di sistem kedepannya perlu ditingkatkan terkait kapasitas memori dan penyimpanan pada server yang dipakai, dan kedepannya diharapkan aplikasi ini disediakan API agar selanjutnya bisa dikembangkan aplikasi berbasis mobile untuk

memudahkan pengguna dalam mengakses aplikasi SKPI.

6. Daftar Pustaka

- [1] M. Pendidikan, D. A. N. Kebudayaan, and R. Indonesia, "Menteri pendidikan dan kebudayaan republik indonesia," pp. 2013–2015, 2013.
- [2] M. P. M. Fadlillah, "Implementasi Kurikulum 2013 Berbasis," p. 13, 2014.
- [3] L. Vinet and A. Zhedanov, "A 'missing' family of classical orthogonal polynomials," *J. Phys. A Math. Theor.*, vol. 44, no. 8, pp. 1689–1699, 2011.
- [4] Pemerintah Pusat, "Peraturan Presiden (PERPRES) tentang Kerangka Kualifikasi Nasional Indonesia," *Peratur. Pres.*, vol., no. 8, pp. 37–39, 2012.
- [5] H. Gore *et al.*, "Django: Web Development Simple & Fast," *Ann. Rom. Soc. Cell Biol.*, vol. 25, no. 6, pp. 4576–4585, 2021.
- [6] R. Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, and W. Ramadhan, "Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 11, no. 1, p. 48, 2021.
- [7] S. Alfeno, D. Rifai, and M. Saepudin, "Utilization of the Django Framework as a Dashboard Model Information System for Raw Material Inventory on PT Bimasakti Karyaprima," *Aptisi Trans. Technopreneursh.*, vol. 1, no. 2, pp. 192–202, 2019.
- [8] L. Setiyani, Y. Rostiani, and T. Ratnasari, "Analisis Kebutuhan Fungsional Sistem Informasi Persediaan Barang Perusahaan General Trading (Studi Kasus : PT. Amco Multitech)," *Owner*, vol. 4, no. 1, p. 288, 2020.
- [9] T. Tumini and H. Septiana, "Penerapan Extreme Programming Dalam Penerapan Perancangan Aplikasi Web Web Food Market," *J. Inf. dan Komput.*, vol. 9, no. 2, pp. 113–121, 2021.
- [10] M. Shalahuddin and R. A. Sukamto, "Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek Edisi Revisi," pp. 25–41, 2018.
- [11] A. Ismunandar, "Peningkatan Kemampuan Pendidik," vol. 6, no. 2, pp. 388–397, 2023.
- [12] J. Panjaitan and A. Pakpahan, "Perancangan Sistem Absensi Care Group Mahasiswa Berbasis Web Menggunakan Metode Agile Development (Studi Kasus: Universitas Advent Indonesia)," *CogITo Smart J.*, vol. 8, no. 2, pp. 321–334, 2022.
- [13] H. Naufal and A. G. Persada, "Desain Interaksi Berbasis User Experience pada Mobile Application : Suatu Tinjauan Literatur," *Automata*, vol. 1, no. 2, pp. 1–5, 2020.
- [14] Houghton and Vivienne, "Moqups: an easy way to create and share mockup designs online without having to know code.," *Univ. Color. Strauss Heal. Sci. Libr.*, vol. 16, 2014.
- [15] D. Selviani, "Pengaruh Penerapan Anggaran Berbasis Kinerja Terhadap Akuntabilitas Kinerja Instansi Pemerintah Kota Cimahi," *L. J.*, vol. 1, no. 2, pp. 117–124, 2021.
- [16] A. Fitriadi and H. A. Tawakal, "Jurnal Informatika Terpadu," *J. Inform. Terpadu*, vol. 7, no. 2, pp. 62–69, 2021.
- [17] M. A. Al Thareq, E. N. Alam, and N. I. Utama, "SEIKO : Journal of Management & Business Pembangunan Website Manajemen Inventori Bahan Baku Makanan Pada Rumah Makan Alam Mutiara Menggunakan Metode RAD (Rapid Application Development)," *SEIKO J. Manag. Bus.*, vol. 6, no. 1, pp. 144–155, 2023.
- [18] T. Dpm, Y. Religia, S. Kom, and M. Kom, "DOKUMEN SISTEM PENJAMINAN MUTU INTERNAL (SPMI) UPB Direktorat Penjaminan Mutu Universitas Pelita Bangsa 2021," 2021.
- [19] S. Atma, Y. D., Idris, N. Bin, &

- Literatur, “Rancang Bangun Sistem PPDB Online SMP-IT AS ’ ADIYAH,” *J. Inform. dan Teknol. Inf.*, vol. 1, no. 1, pp. 25–30, 2023.
- [20] A. Andhini, I. Ibrahim, and Y. Saragih, “Implementasi Aplikasi Styins Home pada Smart Home Security Menggunakan Real-Time Database Firebase,” *J. Ecotipe (Electronic, Control. Telecommun. Information, Power Eng.*, vol. 7, no. 2, pp. 117–126, Oct. 2020.
- [21] A. Ramdhani and A. M. Thantawi, “Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web Dan Dashboard Visualisasi Data Untuk Monitoring Minat Baca Pada SMK Negeri 21,” *IKRA-ITH Inform. J. Komput. dan Inform.*, vol. 8, no. 2, pp. 191–199, 2024.