DESAIN PERANCANGAN APLIKASI MANAJEMEN KOS BERBASIS WEB PADA KOS 2A

Ie Chen [1)] Tony [2)]

[1) 2)] Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Tarumanagara
Jl. Letjen S. Parman No.1, Jakarta Barat 11440 Indonesia
email : iechen.825220057@stu.untar.ac.id[1)], tony@fti.untar.ac.id[2)]

ABSTRAK

Sebagai salah satu rumah kos yang telah beroperasi hampir 15 tahun, Kos 2A di Jakarta Barat masih mengelola data penyewa, pembayaran, serta penanganan keluhan secara manual. Kondisi ini menimbulkan beberapa tantangan, seperti rendahnya efisiensi, kesulitan dalam pengelolaan data, keterlambatan penanganan keluhan, serta pemantauan pembayaran yang kurang optimal.Untuk mengatasi permasalahan tersebut, penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi manajemen rumah kos berbasis web yang mampu mengintegrasikan pengelolaan data penyewa, proses pembayaran, dan penanganan keluhan dalam satu platform terpusat. Integrasi ini diharapkan dapat meningkatkan efisiensi operasional, transparansi, serta kenyamanan bagi pemilik kos maupun para penyewa. Penelitian ini menggunakan metodologi hybrid yang menggabungkan pendekatan Waterfall dan Agile. Tahap perencanaan dan analisis dilakukan secara sistematis mengikuti model Waterfall, sedangkan tahap pengembangan dan pengujian dilakukan secara iteratif berdasarkan prinsip Agile. Melalui pendekatan tersebut, aplikasi yang dihasilkan diharapkan lebih adaptif, terstruktur, dan sesuai dengan kebutuhan operasional Kos 2A.

Key words

Boarding House Management, Website, Hybrid Agile-Waterfall, Operational Digitalization.

1. Pendahuluan

Sebagai ibu kota negara Indonesia, kota Jakarta memiliki jumlah kos-kosan yang sangat banyak untuk memenuhi kebutuhan tempat tinggal para perantau. Rumah kos, yang sering disebut sebagai kos-kosan, merupakan salah satu kebutuhan penting bagi mahasiswa, pekerja, maupun masyarakat umum yang membutuhkan tempat tinggal sementara sebagai solusi hunian selama berada jauh dari rumah asal [1]. Keberadaan usaha kos-kosan yang dibangun oleh masyarakat di sekitar kampus, area perkantoran, dan pusat aktivitas lainnya memberikan kemudahan bagi mahasiswa dan pekerja dari luar daerah yang membutuhkan tempat tinggal [2]. Tersedia berbagai jenis kos, mulai dari yang sederhana hingga yang bersifat eksklusif, yang dapat dipilih sesuai dengan kebutuhan dan kemampuan finansial masing-masing individu [3].

Saat ini Kos 2A masih melakukan operasional pencatatan data penghuni, pengelolaan pembayaran, serta penanganan keluhan secara manual. Dikarenakan masih banyaknya aktivitas yang dilakukan secara manual, seringkali mengurangi efisiensi serta memerlukan waktu, tenaga, dan juga sumber daya manusia. Kendala yang dialami mengganggu dan mengurangi kenyaman pagi penghuni kos dan juga pemilik kos. Pemilik kos kesulitan untuk mengolah berbagai macam data penghuni untuk setiap kamarnya, menerima berbagai macam keluhan dari setiap penghuni melalui aplikasi *Whatsapp.* Banyaknya pesan yang masuk, sering kali membuat pemilik kos melewatkan keluhan dari setiap penghuni yang ada secara tidak disengaja. Kesulitan bagi pemilik untuk melakukan pengecekan pembayaran yang masuk dari setiap penghuni kos juga menjadi kendala bagi operasional

Pemanfaatan teknologi informasi merupakan salah satu solusi yang potensial untuk meningkatkan efisiensi sekaligus efektivitas dalam pengelolaan kos-kosan. Dengan adanya digitalisasi, proses manajemen yang sebelumnya dilakukan secara manual dapat diotomatisasi dan disajikan secara lebih terorganisir. Oleh karena itu, dibutuhkan sebuah perancangan aplikasi manajemen kos-kosan berbasis *web* yang mampu membantu pemilik kos dalam mengelola data para penghuni dengan lebih rapi, sistematis, dan mudah diakses kapan saja. Selain itu, aplikasi ini juga diharapkan dapat menampung dan mendokumentasikan setiap keluhan dari penghuni kos sehingga pemilik dapat memahami permasalahan yang dialami secara lebih terstruktur serta merespons dengan lebih cepat.

Tidak hanya itu, aplikasi manajemen kos ini juga dirancang untuk menyediakan fitur pencatatan pembayaran sewa yang terdokumentasi dengan baik, sehingga baik pemilik maupun penghuni dapat memperoleh transparansi terkait transaksi. Secara garis besar, tujuan dari perancangan aplikasi ini adalah untuk memastikan bahwa setiap informasi yang berkaitan dengan data penghuni, keluhan, serta pembayaran dapat dikelola secara terintegrasi dalam satu *platform* digital. Dengan demikian, diharapkan tercipta sistem pengelolaan

kos yang lebih efisien, transparan, dan memberikan kenyamanan bagi kedua belah pihak, baik pemilik maupun penghuni kos.

2. Dasar Teoritik dan Metodologi

2.1. Dasar Teori

2.1.1 *Website*

Website merupakan sekumpulan halaman yang tersedia di internet dan dapat diakses melalui *browser* untuk berbagai tujuan, seperti memperoleh informasi, melakukan transaksi belanja, berinteraksi sosial, hingga menikmati hiburan. Berdasarkan karakteristiknya, *website* terbagi menjadi dua tipe utama, yaitu statis dan dinamis. *Website* statis menampilkan konten yang bersifat tetap dan hanya dapat diperbarui secara manual, umumnya dibangun menggunakan *HTML* serta *CSS*. Sebaliknya, *website* dinamis memungkinkan adanya interaksi antara pengguna dengan server sehingga isi halaman dapat menyesuaikan kebutuhan, biasanya dikembangkan menggunakan bahasa pemrograman seperti *PHP*, *Python*, atau *JavaScript* [4]. Selain itu, rancangan dan fitur yang terdapat pada sebuah *website* sangat berpengaruh terhadap kemudahan penggunaan serta manfaat yang diperoleh. Hal tersebut secara langsung mendukung fungsi *website* sebagai media promosi sekaligus sarana penyedia informasi digital yang lebih efektif, informatif, dan menarik [5].

2.1.2 Kos

Hunian kos, atau yang lebih dikenal dengan istilah kos-kosan, menjadi salah satu kebutuhan pokok bagi mahasiswa maupun pekerja yang merantau untuk melanjutkan pendidikan atau bekerja jauh dari daerah asal mereka. Bagi para perantau, kos berfungsi sebagai tempat tinggal sementara yang sangat dibutuhkan [6]. Kos sendiri merupakan jenis hunian sederhana yang disewakan kepada masyarakat sebagai tempat tinggal sementara, dengan sistem pembayaran yang biasanya dilakukan secara bulanan ataupun tahunan berdasarkan kesepakatan antara pemilik dan penyewa [7].

2.1.3 Model *Hybrid Agile-Waterfall*

Model *Agile-Waterfall Hybrid* merupakan kombinasi dari dua metode, yakni *Waterfall* dan *Agile*. Pendekatan *Waterfall* biasanya digunakan pada bagian proyek yang membutuhkan kepastian dan struktur yang jelas, seperti regulasi maupun dokumentasi formal, sedangkan *Agile* lebih sesuai untuk aspek yang membutuhkan fleksibilitas, seperti pengembangan fitur dan interaksi dengan pengguna. Penerapan model ini umumnya mencakup tiga tahap, yaitu perencanaan serta analisis dengan pendekatan *Waterfall* untuk menetapkan lingkup proyek dan kebutuhan bisnis sekaligus menyusun dokumentasi, pengembangan dan pengujian dengan prinsip *Agile* melalui siklus iteratif yang cepat, serta tahap integrasi dan implementasi kembali dengan pendekatan *Waterfall* yang menekankan pada pengujian menyeluruh, pembuatan dokumentasi akhir, dan penerapan sistem. Model ini dianggap ideal bagi proyek yang memiliki regulasi ketat namun tetap memerlukan fleksibilitas *Agile*, meskipun tantangan utamanya adalah menjaga keseimbangan antara kontrol ketat khas *Waterfall* dengan kemampuan adaptif dari *Agile* tanpa mengurangi produktivitas tim [8].

2.2. Metodologi

2.2.1 *Software Development Lifecycle (SDLC*)

Software Development Life Cycle (SDLC) merupakan kerangka kerja yang berfungsi sebagai landasan utama dalam proses rekayasa perangkat lunak, karena di dalamnya terdapat pendekatan yang sistematis, terukur, dan terstruktur untuk menghasilkan perangkat lunak yang berkualitas tinggi. Model ini tidak hanya berperan sebagai pedoman dalam merancang, membangun, dan menguji perangkat lunak, tetapi juga menjadi acuan dalam memastikan bahwa setiap tahap pengembangan berjalan sesuai dengan standar yang telah ditetapkan. Dalam konteks perkembangan teknologi yang semakin cepat serta meningkatnya kebutuhan pengguna terhadap sistem yang lebih efisien, pemilihan model *SDLC* yang tepat menjadi faktor penentu keberhasilan sebuah proyek. Model yang sesuai akan mampu meningkatkan efisiensi pengelolaan waktu, biaya, serta sumber daya, sekaligus menjaga efektivitas kerja tim [9].

Seiring berjalannya waktu, berbagai metodologi, teknik, serta alat bantu dalam bidang rekayasa perangkat lunak telah diciptakan dan disempurnakan dengan tujuan untuk mempercepat proses pengembangan, mengurangi biaya produksi, dan mempersingkat waktu penyelesaian, terutama pada sistem yang bersifat kompleks [16]. Selain itu, inovasi-inovasi dalam metodologi *SDLC* juga diarahkan untuk mendukung terciptanya perangkat lunak yang lebih adaptif terhadap perubahan kebutuhan pengguna dan lebih mampu menghadapi tantangan dalam lingkungan teknologi yang dinamis. Meskipun terdapat banyak model dan pendekatan yang dapat digunakan, setiap tahapan utama dalam *SDLC* mulai dari perencanaan, analisis, desain, implementasi, pengujian, hingga pemeliharaan memiliki peran yang sangat vital dan tidak dapat diabaikan. Oleh karena itu, penerapan yang konsisten pada setiap fase sangat diperlukan agar kualitas perangkat lunak yang dihasilkan tidak hanya memenuhi standar fungsional, tetapi juga dapat memberikan nilai tambah bagi penggunanya [10].

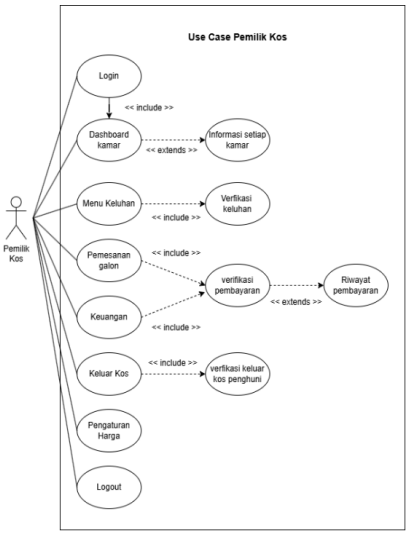
3. Hasil Percobaan

Perancangan aplikasi ini mencakup pembangunan satu website utama yang dapat digunakan oleh penghuni maupun pemilik kos. Untuk penghuni, *website*

menyediakan fitur utama seperti pembayaran sewa, manajemen informasi kamar, pelaporan keluhan, serta pengajuan keluar kos. Sementara itu, pemilik kos memperoleh akses tambahan berupa menu keuangan yang berfungsi untuk memantau secara detail seluruh transaksi dan aktivitas pengelolaan keuangan. Dengan adanya fitur ini, pemilik kos dapat lebih mudah mengontrol arus kas dan memastikan transparansi dalam pengelolaan kos. Dalam proses perancangannya, *Unified Modeling Language* (UML) digunakan sebagai alat bantu pemodelan agar rancangan sistem dapat tersusun secara terstruktur, terdokumentasi dengan baik, serta mudah dipahami oleh pihak terkait dalam tahap pengembangan selanjutnya.

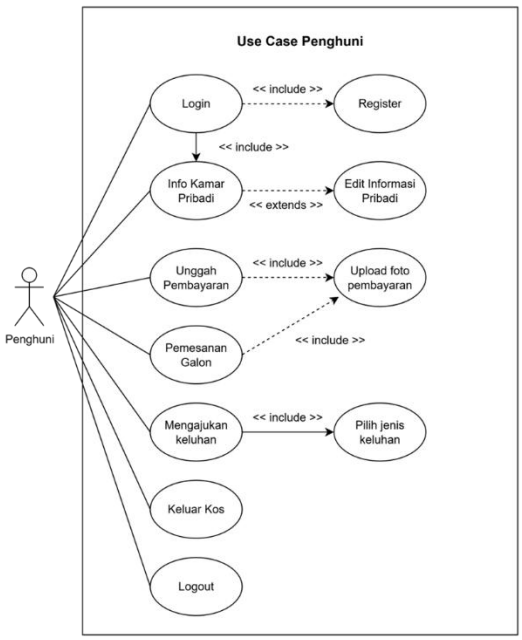
3.1 *Use Case* Diagram

Use case diagram digunakan untuk memodelkan perilaku suatu sistem dengan menampilkan alur tindakan atau skenario penggunaan yang melibatkan interaksi antara sistem dengan aktor, yaitu pengguna eksternal atau pihak yang berhubungan langsung dengan sistem. Diagram ini berfungsi untuk memperlihatkan hubungan serta kolaborasi antara aktor dan sistem dalam menjalankan fungsi atau layanan tertentu [11]. Tujuan utama dari pembuatan *use case* diagram adalah memberikan kemudahan bagi analis dalam merancang serta menyusun kebutuhan yang diperlukan dalam proses pengembangan sistem [12]. Adapun rancangan *use case* pada sistem manajemen kos yang menjelaskan setiap langkah aktivitas yang dapat dilakukan oleh pemilik kos seperti: *login*, *dashboard* kamar, menu keluhan, pemesanan galon, dan keuangan. *Use case* dapat dilihat pada **Gambar 1**,



Gambar 1. Use Case Pemilik Kos

Rancangan *use case* lain juga disusun untuk sistem manajemen kos, yang berfungsi menjelaskan secara lebih rinci setiap langkah aktivitas yang dapat dilakukan oleh penghuni. Beberapa aktivitas yang tercakup di dalamnya meliputi proses registrasi akun, masuk ke dalam sistem, akses terhadap informasi kamar untuk melihat informasi pribadi, penggunaan menu keluhan untuk menyampaikan permasalahan, hingga melakukan pemesanan kebutuhan tambahan seperti galon. Dengan adanya *use case* ini, alur interaksi penghuni dengan sistem dapat digambarkan secara jelas dan terstruktur sehingga memudahkan dalam memahami fungsionalitas yang tersedia. Rancangan *use case* tersebut dapat dilihat pada **Gambar 2.**



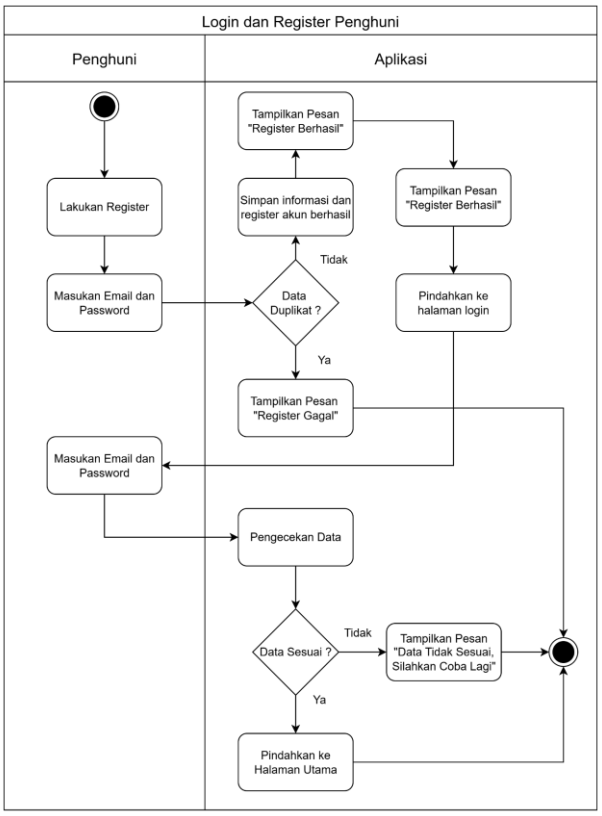
Gambar 2. Use Case Penghuni

3.2 *Activity* Diagram

Activity diagram adalah salah satu jenis diagram perilaku dalam *Unified Modeling Language* (UML) yang digunakan untuk merepresentasikan aspek dinamis dari suatu sistem. Diagram ini dapat dianggap sebagai penyempurnaan dari *flowchart*, karena tidak hanya memodelkan urutan aktivitas, tetapi juga menggambarkan aliran kendali, percabangan keputusan, serta parallelisme yang terjadi di dalam sistem. Dengan memanfaatkan *activity* diagram, logika prosedural maupun alur proses bisnis dapat ditunjukkan secara lebih sistematis dan terperinci.

Selain itu, *activity* diagram berperan penting dalam memberikan visualisasi yang jelas mengenai bagaimana suatu proses dimulai, dijalankan, hingga berakhir. Diagram ini juga membantu menganalisis interaksi antar aktivitas, mengidentifikasi potensi hambatan dalam alur kerja, serta memperlihatkan skenario alternatif yang mungkin terjadi selama proses berlangsung. Oleh karena itu, *activity* diagram tidak hanya bermanfaat dalam konteks pengembangan perangkat lunak, tetapi juga sangat relevan untuk memodelkan proses bisnis non-

teknis, sehingga mampu menjadi alat bantu komunikasi yang efektif antara analis sistem, pengembang, maupun pemangku kepentingan lainnya [13]. *Acitivity* Diagram dapat dilihat pada **Gambar 3**.

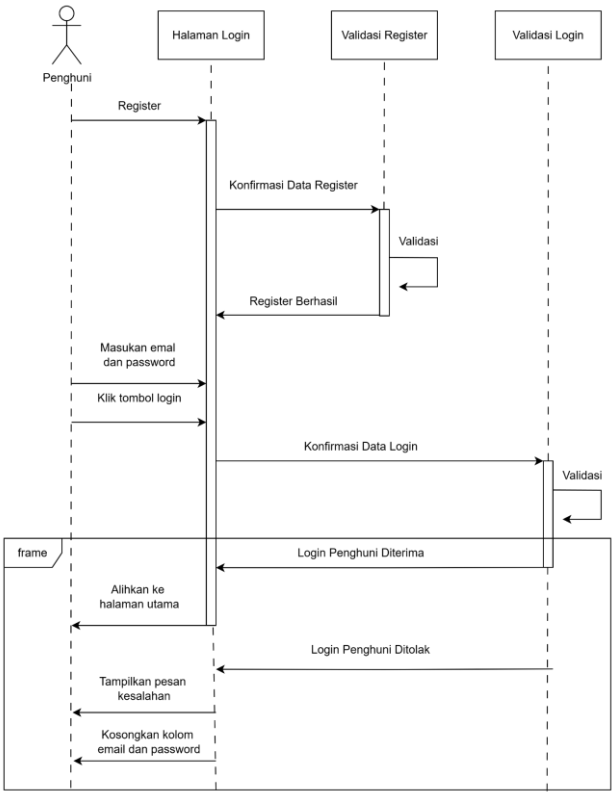


Gambar 3. *Activity* Diagram

3.3 *Sequence Diagram*

Sequence diagram merupakan salah satu representasi visual dalam *UML* yang digunakan untuk memperlihatkan interaksi antar objek yang berada di dalam maupun di sekitar sistem, termasuk aktor, antarmuka (*display*), serta komponen lainnya. Interaksi tersebut digambarkan dalam bentuk pesan (*message*) yang disusun berdasarkan urutan waktu. Diagram ini memiliki dua dimensi utama, yaitu dimensi vertikal yang menggambarkan alur waktu, serta dimensi horizontal yang menunjukkan objek-objek yang berhubungan satu sama lain.

Dalam penggunaannya, *sequence* diagram berfungsi untuk memodelkan *use case scenario*, yaitu serangkaian langkah yang dilakukan sistem sebagai respons terhadap suatu peristiwa (*event*) hingga menghasilkan keluaran tertentu. Proses pemodelan biasanya diawali dengan mengidentifikasi pemicu aktivitas, kemudian dilanjutkan dengan menggambarkan alur interaksi, perubahan status, serta hasil yang muncul dalam sistem. Setiap objek, termasuk aktor eksternal, digambarkan memiliki *lifeline* berbentuk garis vertikal, sedangkan pesan antar objek divisualisasikan menggunakan panah yang menunjukkan arah komunikasi. Dengan demikian, *sequence* diagram membantu dalam memahami secara detail urutan interaksi yang terjadi pada suatu sistem [14]. *Sequence* Diagram dapat dilihat pada **Gambar 4**.



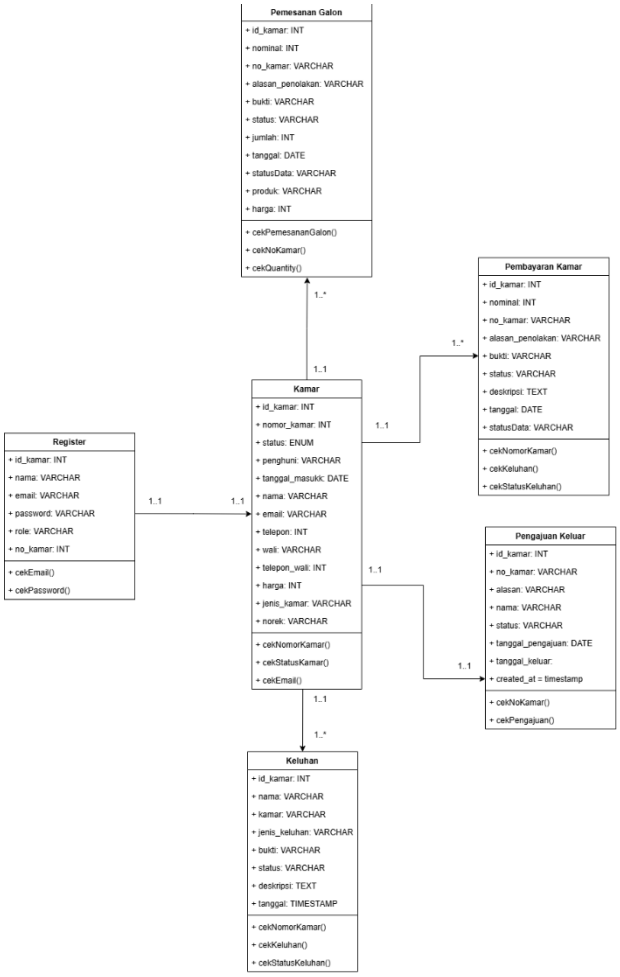
Gambar 4. *Sequence* Diagram

3.4 *Class Diagram*

Setelah *use case* dijelaskan melalui activity diagram, tahap berikutnya adalah merancang struktur program yang akan menjalankan proses tersebut. Dalam *UML*, struktur ini didokumentasikan menggunakan *class* diagram yang menggambarkan kumpulan kelas beserta relasi antar kelas. *Class* diagram memiliki keterkaitan erat dengan implementasi nyata pada kode program, sehingga kerap dijadikan acuan dalam tahap pengembangan perangkat lunak. Beberapa perangkat lunak perancangan *UML* bahkan menyediakan fitur untuk menghasilkan class diagram langsung dari kode sumber yang sudah ada melalui proses *reverse engineering*. Sebaliknya, terdapat pula metode *forward engineering*, yaitu pembuatan kode sumber secara otomatis dari *class* diagram yang telah dirancang [15].

Dengan kemampuannya menampilkan hubungan antar kelas, atribut, metode, serta keterkaitan logis antar objek, *class* diagram tidak hanya berfungsi sebagai dokumentasi statis, tetapi juga sebagai alat bantu untuk memahami struktur sistem secara lebih mendalam. Diagram ini mempermudah analisis dan perancangan fungsi-fungsi yang akan diimplementasikan, serta membantu pengembang mengontrol alur komunikasi antar objek secara kronologis, sehingga sistem dapat

berjalan sesuai kebutuhan yang telah ditetapkan. *Class* Diagram dapat dilihat pada **Gambar 5**.



Gambar 5. *Class* Diagram

3.5 *Wireframe Design*

Pada awalnya, sebagian desainer beranggapan bahwa *wireframe* tidak begitu diperlukan karena dianggap memakan waktu dan kurang memberikan produktivitas. Namun, seiring berkembangnya teknologi dan hadirnya berbagai alat desain modern, pandangan tersebut mengalami perubahan. Saat ini, *wireframe* dapat diperkaya dengan interaktivitas sederhana sehingga menghasilkan *prototype* dinamis berfidelitas rendah. *Prototype* semacam ini membantu memvisualisasikan bagaimana fungsi produk akhir akan berjalan, menjadikan *wireframe* jauh lebih bermanfaat dibandingkan sekadar sketsa statis. *Wireframe* sendiri dapat diibaratkan sebagai rancangan awal dari sebuah aplikasi atau halaman *web*, yang menampilkan struktur, tata letak dasar, serta alur informasi tanpa menonjolkan detail visual. Dalam proses desain pengalaman pengguna (UX), *wireframe* memegang peranan penting karena mampu menyatukan pemahaman seluruh anggota tim mengenai rencana yang disusun. Dengan demikian, *wireframe* berfungsi sebagai fondasi yang kuat untuk mendukung keberhasilan suatu proyek [16]. *Wireframe Design* dapat dilihat pada **Gambar 6.**

Gambar 6. *Wireframe Design*

4. Kesimpulan

Berdasarkan hasil analisis, perancangan, dan pembahasan yang telah dilakukan dalam penelitian ini, dapat disimpulkan bahwa pengembangan aplikasi manajemen kos berbasis *web* pada Kos 2A mampu memberikan solusi terhadap berbagai permasalahan operasional yang sebelumnya dilakukan secara manual. Adapun beberapa kesimpulan yang dapat diambil adalah sebagai berikut:

1. Mengintegrasikan pengelolaan data penghuni, pembayaran, dan keluhan dalam satu sistem terpusat, sehingga meningkatkan efisiensi operasional.

2. Penerapan model pengembangan *Hybrid Agile-Waterfall* untuk proses perancangan yang lebih adaptif dan terstruktur.

3. Pemanfaatan metode pemodelan seperti *UML* dan *wireframe design* untuk membantu dalam menyusun rancangan sistem yang terorganisir dan mudah dipahami.

REFERENSI

[1] R. Setiawan, A. D. Supriatna, and A. H. Kusuma, "Rancang Bangun Sistem Informasi Pengelolaan Rumah Kos Deo Garut Berbasis Web," *Jurnal Algoritma*, vol. 17, no. 2, pp. 368–377, 2020.

[2] J. Ismail, "Activity Based Costing (ABC) pada kos-kosan," Jurnal Akuntansi STIE Muhammadiyah Palopo, vol. 7, no. 1, 2021.

[3] E. D. Dhano, F. L. Banda, and S. Kapa, "Faktor-Faktor yang Mempengaruhi Tingkat Kepatuhan Pemilik Kos dalam Membayar Pajak Rumah Kos: Studi Kasus pada Pemilik Usaha Kos-Kosan di Kota Kupang," Jurnal Riset Ilmu Akuntansi, vol. 1, no. 2, pp. 15–23, 2021.

[4] E. Hariadi and A. Rosyidi, "Pengaruh desain/fitur pada kemanfaatan dan kemudahan akses website bisnis terhadap penerimaan pelanggan di Kabupaten Klaten Provinsi Jawa Tengah," Innovative: Journal of Social Science Research, vol. 5, no. 1, pp. 3513–3526, 2025.

[5] Y. Wahyudin and D. N. Rahayu, "Analisis metode pengembangan sistem informasi berbasis website: a literature review," Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi, vol. 15, no. 3, pp. 119–133, 2020.
[6] R. I. D. H. O. T. U. L. Aslam K, "Pengaruh usaha kos terhadap tingkat pendapatan usaha mikro kecil dan menengah di Kecamatan Bara," Doctoral dissertation, Institut Agama Islam Negeri Palopo, 2023.
[7] O. Arifin, M. Murnawan, M. Pandia, D. O. Sihombing, M. Fahrurrozi, S. Sepriano, and M. Lutfi, *Buku Ajar Pemrograman Web*. Indonesia: PT. Sonpedia Publishing Indonesia, 2023. [Online]. Available: https://books.google.co.id/books?id=q9sKEOAAQBAJ
[8] Osis and U. Donins, Topological UML Modeling: An Improved Approach for Domain Modeling and Software Development. Amsterdam, Netherlands: Elsevier, 2017. [Online]. Available: https://www.google.co.id/books?id=jJLUDQAAQBAJ
[9] S. Pargaonkar, "A comprehensive research analysis of software development life cycle (SDLC) agile & waterfall model advantages, disadvantages, and application suitability in software quality engineering," International Journal of Scientific and Research Publications, vol. 13, no. 8, pp. 120–124, 2023.
[10] C. Dongmo, "A Review of Non-Functional Requirements Analysis Throughout the SDLC," Computers, vol. 13, no. 12, p. 308, 2024. doi: 10.3390/computers13120308
[11] "SLDC Adalah: Pengertian, Tahapan, dan Model," Course-Net. [Online]. Available: https://course-net.com/blog/sldc-adalah
[12] Setiyani, "Desain Sistem: Use Case Diagram," in Prosiding Seminar Nasional Inovasi dan Adopsi Teknologi (INOTEK), vol. 1, no. 1, pp. 246–260, Nov. 2021.
[13] F. Catthoor, T. Basten, N. Zompakis, M. Geilen, and P. G. Kjeldsberg, System-Scenario-based Design Principles and Applications. Cham, Switzerland: Springer Nature, n.d. [Online]. Available: https://www.google.co.id/books?id=7AqwDwAAQBAJ
[14] W. S. Davis and D. C. Yen, Eds., The Information System Consultant's Handbook: Systems Analysis and Design. Boca Raton, FL, USA: CRC Press, n.d. [Online]. Available: https://www.google.co.id/books?id=I628DwAAQBAJ
[15] R. Julihardi, "Sistem pendukung keputusan pemilihan karyawan Trans Semarang berprestasi menggunakan metode Weighted Aggregated Sum Product Assessment (WASPAS)," Tugas Akhir, Univ. Semarang, Semarang, Indonesia, 2023.
[16] A. Fatah, F. A. Mufarroha, and O. M. A. Husnah, "Perancangan antarmuka pengguna sistem informasi akademik berbasis wireframing," SimanteC, vol. 11, no. 1, pp. 97–108, 2022.

Ie Chen, saat ini sebagai mahasiswa program studi Sistem Informasi Universitas Tarumanagara angkatan 2022.

Tony, memperoleh gelar S.Kom. dari Universitas Tarumanagara, Indonesia pada tahun 2005. Gelar M.Kom. dari Universitas Indonesia tahun 2010 dan gelar Ph.D. dari Curtin University of Technology, Australia pada tahun 2021. Saat ini sebagai staf Pengajar Program Studi Sistem Informasi Universitas Tarumanagara.