



JURNAL MANAJEMEN INFORMATIKA

Halaman Jurnal: <http://jurnal.stmik-dci.ac.id/index.php/jumika/>

Halaman LPPM STMIK DCI : <http://lppm.stmik-dci.ac.id/>



PENGEMBANGAN *FRONTEND* UNTUK PERBAIKAN DAN PENINGKATAN *WEBSITE* SALONKITA

Akik Hidayat¹, Ahmad Yahya Salim²

Prodi Teknik Informatika, Universitas Padjadjaran Bandung

Email : akik@unpad.ac.id¹, ahmad20022@mail.unpad.ac.id²

ABSTRAK

Salonkita merupakan perusahaan yang bergerak di industri kecantikan dengan salah satu produk mereka yaitu aplikasi *point of sale* untuk mengelola salon. Dengan meningkatnya teknologi web di zaman modern, perubahan untuk meningkatkan kualitas aplikasi web merupakan sebuah langkah baik dari sudut pandang keberlanjutan. Penelitian pada program penelitian ini bertujuan untuk mengatasi gagasan tersebut. Melalui penelitian ini, *frontend* aplikasi direncanakan sedemikian rupa untuk bisa dikembangkan dan mencapai tujuan yang diharapkan. Keluaran akhir penelitian yang diharapkan adalah perangkat lunak yang mampu menggunakan *framework* pengembangan yang mendukung fitur-fitur web modern beserta dengan tampilan responsif untuk penggunaan yang lebih luas jangkauannya. Hasil dari penelitian adalah antarmuka *frontend* aplikasi yang mampu melakukan koneksi ke aplikasi *backend* dan bersifat responsif. Dari hasil tersebut dapat disimpulkan penelitian mampu mencapai tujuan serta memperoleh nilai kepuasan yang baik.

Kata Kunci: Salonkita, Web, *Frontend*.

I. PENDAHULUAN

Perkembangan sistem informasi yang pesat memicu digitalisasi berbagai macam aktivitas pada masyarakat tidak terkecuali pada lingkungan industri. Fenomena ini dapat terlihat dengan jelas dari berbagai produk teknologi seperti

perangkat lunak yang memfasilitasi aktivitas masyarakat hingga dapat dilakukan melalui perangkat-perangkat elektronik, seperti contoh ojek *online* dan belanja *online*. Selain itu, banyak pula pelaku bisnis dan jasa yang melihat potensi pada perkembangan teknologi ini untuk

bisa mengembangkan suatu produk yang bisa menjadi solusi dari permasalahan di era modern ini.

PT Salon Kita Indonesia atau dikenal juga dengan salonkita merupakan *start-up* yang bergerak dalam bidang digitalisasi bisnis salon. Saat ini, salonkita mengoperasikan 2 (dua) domain yaitu salonkita.id sebagai platform *marketplace* dan salonkita.co.id sebagai *website* resmi perusahaan. Melalui *website* resmi, salonkita memberikan informasi, promosi dan edukasi kepada pelanggan maupun calon pelanggan.

Website salonkita.co.id versi 1 saat ini menggunakan Odoo ERP dengan python sebagai bahasa pemrogramannya dan PostgreSQL sebagai *database*. Sementara itu, infrastruktur pendukung menggunakan Cloud PVS yang ada di AWS *Singapore* dengan sistem operasi Linux Ubuntu 22.04 LTS dan aaPanel sebagai server *management*. Dari sisi keamanan dan pengaturan domain sementara ini menggunakan CloudFlare.

Sejauh ini, versi pertama *website* salonkita.co.id sudah memberikan fondasi yang baik namun sistem

backend terasa kaku dan kompleks. Selain itu, ada beberapa hal yang perlu ditingkatkan seperti dwibahasa, UI/UX, pengelolaan konten, pengelolaan pelanggan dan pemesanan. Oleh karena itu, diputuskan bahwa ada kebutuhan untuk mendesain ulang *website* baik

secara teknologi, UI/UX *frontend* dan *backend*.

Metode *Waterfall* dipilih sebagai metode pengembangan aplikasi ini karena perancangan aplikasi yang sudah terdefiniskan. Metode tersebut juga diharapkan dapat mencapai tujuan awal tanpa distraksi dari kebutuhan lain.

II. Tinjauan Pustaka

2.1 Metode Pengembangan *Waterfall*

Model proses *Waterfall*, atau biasa disebut *classic life cycle*, menggunakan pendekatan yang sistematis dan berurutan dalam pengembangan perangkat lunak. Proses ini dimulai dari spesifikasi kebutuhan pengguna melalui perencanaan, pemodelan, konstruksi, serta penerapan, dan diakhiri dengan dukungan berkelanjutan terhadap perangkat lunak yang telah selesai (Pressman, 2009). Setiap fase dalam model *Waterfall* harus diselesaikan secara berurutan sebelum berpindah ke fase berikutnya

2.2 *Web Responsif*

Web responsif merupakan serangkaian teknik dalam pengembangan situs web dengan tujuan dapat menyesuaikan dirinya sendiri saat digunakan oleh berbagai perangkat dan mampu membentuk kembali dirinya sendiri pada berbagai ukuran layar dan resolusi (Wasino et al., 2023). Esensinya adalah adanya sifat adaptif dari situs web itu sendiri yang dapat menyesuaikan dengan ukuran dari klien web konsumen layanan sistem tanpa harus

meningkatkan kompleksitas pengembangan dan biaya produksi.

2.3 *JavaScript*

JavaScript merupakan bahasa *script* populer yang dipakai untuk menciptakan halaman web yang dapat berinteraksi dengan pengguna dan dapat merespons *event* yang terjadi pada halaman. JavaScript merupakan perekat yang menyatukan halaman-halaman web. Akan sangat susah menjangkau halaman web komersial yang tidak memuat kode JavaScript.

Meskipun JavaScript secara umum digunakan sebagai bahasa *script* sisi-klien, ia dapat pula digunakan (pada konteks lain) selain pada browser web. Netscape menciptakan JavaScript sisi- server yang bisa diprogram sebagai bahasa CGI, seperti Python atau Perl (Siahaan & Sianipar, 2018).

2.4 *TypeScript*

TypeScript adalah ekstensi dari JavaScript yang ditujukan untuk menangani kekurangannya dalam mengembangkan dan memelihara aplikasi skala besar. Secara sintaksis, TypeScript merupakan superset dari EcmaScript 5, sehingga seluruh program JavaScript adalah program TypeScript pula. TypeScript memperkaya JavaScript dengan sistem modul, kelas, antarmuka, dan sistem tipe statis. Karena TypeScript ditujukan untuk menyediakan sebuah bentuk bantuan kepada *programmer*, sistem modul dan sistem tipe dari TypeScript bersifat fleksibel dan mudah

digunakan. Secara khusus, mereka mendukung banyak praktik pemrograman JavaScript umum. Mereka juga memungkinkan adanya *tooling* dan pengalaman IDE yang sebelumnya diasosiasikan dengan bahasa seperti C dan Java. Contohnya saja, sistem tipe membantu untuk menemukan kesalahan secara statis dan memungkinkan dukungan lain untuk pengembangan program (seperti contoh, menyarankan metode apa yang bisa dipanggil pada suatu objek). Dukungan untuk kelas-kelas ini sejajar dengan proposal yang sekarang sedang distandardisasikan untuk EcmaScript 6 (Bierman et al., 2014).

2.5 *UML*

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang dominan dalam pengembangan perangkat lunak berorientasi objek. Bahasa ini diadopsi oleh *Object Management Group* (OMG) sebagai bahasa pemodelan standar pada tahun 1997. UML 1.x mendefinisikan sembilan teknik diagram: diagram kelas, diagram objek, diagram komponen, diagram penyebaran, diagram penggunaan status, diagram status, diagram aktivitas, diagram urutan, dan diagram kolaborasi (Booch et al., 1999).

Munculnya UML merupakan tonggak utama dalam bidang sistem informasi sehingga para peneliti dan praktisi dapat fokus untuk meningkatkan dan meningkatkan pemodelan tunggal bahasa alih-alih

beberapa lusin pemodelan (Siau, 2007).

Berikut merupakan beberapa jenis UML:

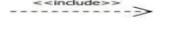
1. Use Case Diagram

Diagram use case adalah jenis diagram UML (*Unified Modeling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktornya (Booch et al., 1999). Sebuah Use case dapat menggambarkan jenis interaksi antara pengguna sistem dan sistem. Use case adalah sesuatu yang mudah dipelajari.

Di dalam Use Case Diagram terdapat simbol-simbol yang menjelaskan pada setiap bagian. Simbol-simbol pada Use Case Diagram berfungsi untuk membangun use case sesuai dengan sistem yang akan dibuat oleh developer.

Tabel 2.1 Use Case Diagram.

No	Simbol	Keterangan
1		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
2		Orang atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. Biasanya dinyatakan menggunakan kata benda atau nama aktor.
3		Komunikasi antara aktor dengan use case yang berpartisipasi pada diagram

4		Relasi use case tambahan ke sebuah use case, dimana use case yang ditambahkan dapat berdiri sendiri meski tanpa use case tambahan itu. Arah panah mengarah pada use case yang ditambahkan.
5		Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		Relasi use case tambahan ke sebuah use case, dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Arah panah include mengarah pada use case yang dibutuhkan.

2. Activity Diagram

Activity Diagram atau dalam bahasa Indonesia berarti diagram aktivitas, yaitu diagram yang dapat memodelkan berbagai proses yang terjadi pada sistem. Urutan operasi sistem operasi seperti itu dijelaskan secara vertikal (Booch et al., 1999). Activity Diagram merupakan salah satu contoh diagram UML dalam pengembangan usecase

Tabel 2.2 Activity Diagram

No	Simbol	Keterangan
1		Status awal sebagai penanda tindakan awal atau titik awal untuk sebuah Activity Diagram.
2		Aktivitas yang ada atau dilakukan suatu sistem.
3		Percabangan diperlukan jika ada beberapa pilihan aktivitas atau jalan alternatif.
4		Penggabungan flow yang semula dipecah menjadi beberapa bagian.
5		Status akhir sebagai penanda bahwa suatu proses sudah berakhir.

6	Swimlane 	Swimlane memisahkan activity diagram menjadi kolom dan baris untuk membagi tanggung jawab objek-objek yang melakukan suatu aktivitas.
---	---	---

III. ANALISIS DAN PERANCANGAN

3.1 Deskripsi Aplikasi

Aplikasi web *frontend* salonkita ditujukan sebagai wajah utama dari perusahaan untuk memperkenalkan profil perusahaan kepada masyarakat yang dapat diakses melalui internet. Melalui aplikasi ini juga berita-berita serta blog bacaan mengenai perusahaan dan hal seputar kecantikan dapat dibaca yang dapat meningkatkan pengetahuan khalayak mengenai apa yang dilakukan oleh salonkita.

3.2 Tahap Komunikasi dan Perencanaan

Tahap komunikasi terdiri atas inisiasi penelitian dan analisis kebutuhan penelitian. Inisiasi penelitian dilakukan dengan menguraikan tujuan dan ruang lingkup penelitian. Langkah selanjutnya adalah melakukan analisis terhadap kebutuhan penelitian, termasuk kebutuhan sistem pengguna. Proses analisis kebutuhan ini sangat penting untuk memastikan bahwa semua aspek penelitian dipertimbangkan dengan baik sebelum melangkah lebih jauh dalam pengembangannya.

1. Kebutuhan Sistem

Berdasarkan analisis dari tim pengembang, sistem akan dibentuk berdasarkan API yang nanti akan disediakan dari *backend* aplikasi tersebut. Tabel 3.1 menjelaskan tentang kebutuhan sistem yang direncanakan.

Tabel 3.1 Kebutuhan Sistem Aplikasi

No	Nama	Deskripsi
1.	Halaman Utama	Sebagai pengguna, saya dapat mengakses halaman utama dan mendapatkan informasi singkat mengenai web dan dapat melakukan navigasi ke halaman lainnya.
2	<i>About</i>	Sebagai pengguna, saya dapat mengakses halaman <i>about</i> yang berisi informasi tentang perusahaan.
3	<i>News</i>	Sebagai pengguna, saya ingin melihat berita terbaru dari web aplikasi baik berupa kabar perusahaan ataupun blog bacaan terbaru seputar kecantikan.
4	<i>Blog</i>	Sebagai pengguna, saya ingin melihat pilihan blog seputar salon dan tip kecantikan.
5	<i>Detail Blog</i>	Sebagai pengguna, saya dapat mengakses halaman yang berisi detail lebih lanjut dari blog.

2. User Story

Dengan menggunakan kebutuhan sistem sebagai dasar alur penggunaan sistem aplikasi, *user story* diturunkan sebagai hasil dalam tahap mendefinisikan kebutuhan sistem aplikasi yang dibangun. *User story* merupakan jalan untuk mendapatkan tujuan aplikasi beserta kriteria untuk mengetahui apakah sistem berjalan sesuai dengan yang diinginkan saat pengujian.

Tabel 3.2 *User Story* Aplikasi

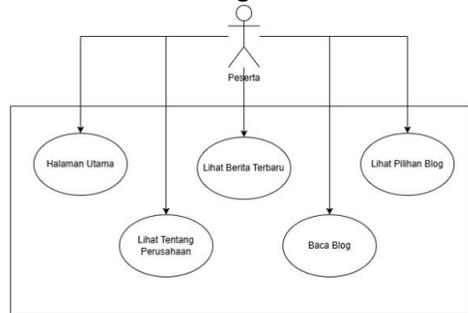
No.	Nama	Acceptance Criteria
1	Halaman Utama	1. Sistem dapat menampilkan halaman utama dengan Deskripsi singkat mengenai aplikasi. 2. Terdapat elemen untuk navigasi ke halaman lain.
2	About	3. Sistem dapat menampilkan halaman tentang perusahaan.
3	News	4. Pengguna dapat melihat halaman berita terbaru. 5. Pengguna dapat mengakses detail blog yang ada pada halaman ini.
4	Blog	6. Sistem dapat menampilkan berbagai pilihan blog untuk dibaca. 7. Pengguna dapat mengakses detail blog yang ada pada halaman ini.

5	Detail Blog	8. Sistem dapat menampilkan halaman berisi blog sesuai <i>slug</i> . 9. Tampilan data blog berupa <i>markdown</i> .
---	-------------	--

3. Desain Diagram Aplikasi

Pada tahap desain ini, manifestasi sistem digambarkan menggunakan diagram-diagram *Unified Modelling Language* (UML) yang bisa memberikan ide akan bagaimana sistem itu terlihat dari sudut pandang logika sistem sampai tampilan antarmuka. Diagram yang ditampilkan dalam tahap desain ini meliputi *use case diagram* dan *activity diagram*.

A. Use Case Diagram



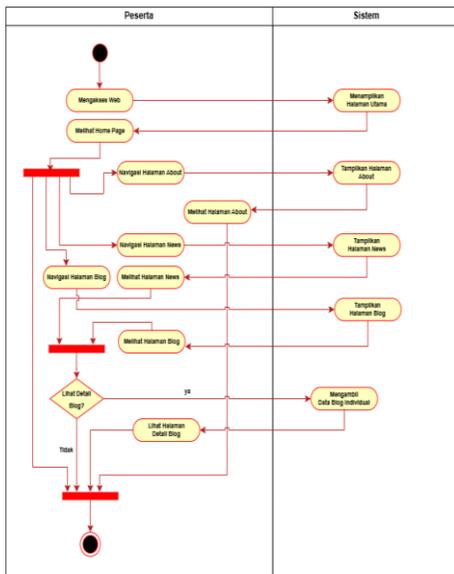
Gambar 3.1 *Use Case Diagram*

Use case diagram di atas menjelaskan kasus penggunaan yang mungkin dilakukan pengunjung aplikasi. Setiap fitur di atas memiliki halaman masing-masing untuk menyajikan fungsi mereka

B. Activity Diagram

Activity diagram menggambarkan aliran penelitian dari suatu sistem. *Activity diagram* aplikasi digambarkan pada dua *swimlane* yang memisahkan aktivitas dari sisi

pengguna dan sistem. Gambar 3.2 menunjukkan alur sistem dalam menangani awal akses pengguna sampai dengan selesainya pengguna mengakses sistem web dari berbagai kasus penggunaan yang memungkinkan.



Gambar 3.2 Activity Diagram

IV. HASIL DAN PEMBAHASAN

4.1 Implementasi program

Implementasi program dibangun dengan fondasi dari seluruh perancangan yang telah dibuat pada bab sebelumnya. Pengembangan tampilan aplikasi ini tidak lupa menggunakan prinsip web responsif agar tampilan dapat digunakan pada berbagai macam perangkat. Implementasi dari aplikasi web salonkita adalah sebagai berikut:

4.1.1 Halaman Utama

Halaman utama merupakan halaman tempat awal pengunjung berada saat mengakses aplikasi ini. Halaman ini menampilkan informasi

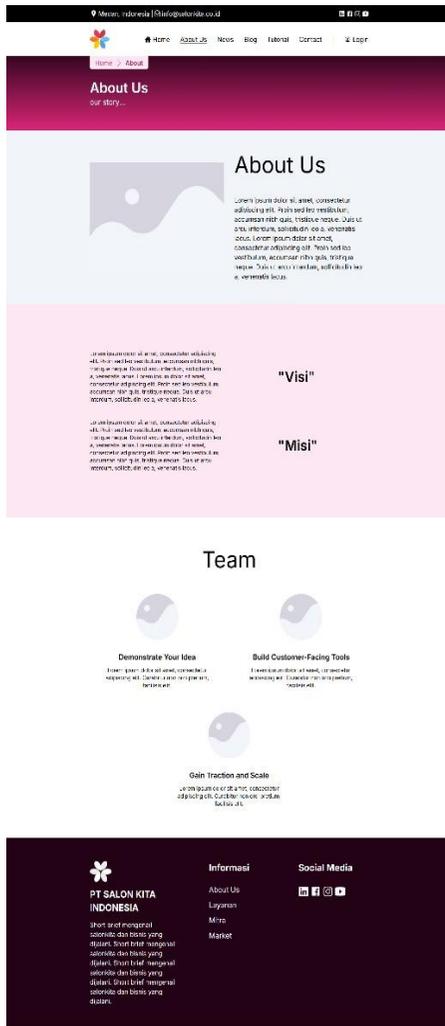
singkat mengenai perusahaan ini data tambahan kabar terbaru dan mitra perusahaan. Tata letak halaman pengguna terdiri dari menu navigasi atau *navigation bar* di atas halaman disusul konten halaman itu sendiri yang kemudian terakhir diikuti dengan konten halaman tersebut dan diakhiri dengan *footer*. Tampilan halaman utama dapat dilihat pada Gambar 4.1 untuk tampilan saat diakses melalui perangkat komputer *desktop*.



Gambar 4.1 Potongan Halaman Utama

4.1.2 Halaman Register

Setelah melihat introduksi dari aplikasi, pengunjung dapat mencoba untuk melakukan navigasi ke halaman lain melalui menu navigasi. Halaman selanjutnya yang dibangun adalah halaman *about*. Pada halaman ini konten bersifat statis karena ketiadaannya data dinamis. Implementasi halaman *about* dapat dilihat pada Gambar 4.2.

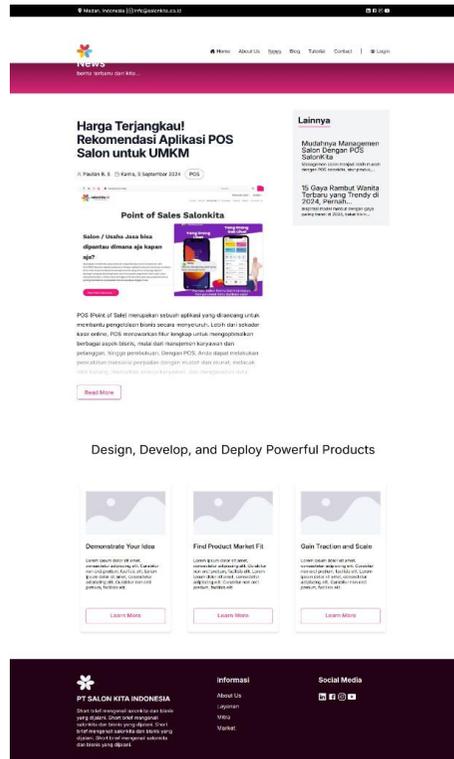


Gambar 4.2 Halaman About

4.1.3 Halaman News

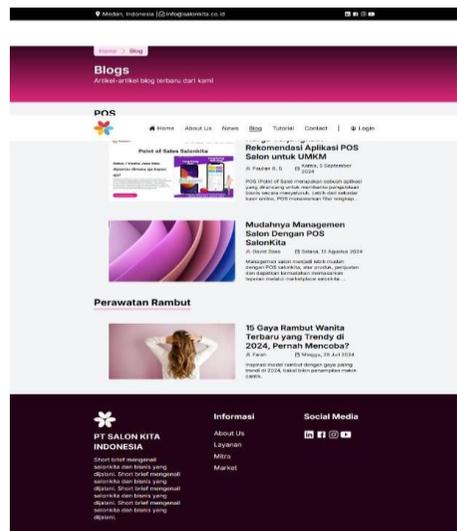
Halaman *news* memberikan pengguna berita-berita terbaru dari perusahaan yang berbentuk blog. Data blog ini diambil berdasarkan waktu terbaru dan hanya berjumlah 10 buah data maksimal. Setelah data berhasil diambil, data blog yang paling baru dibuat akan ditampilkan pada komponen berita disorot. Setiap komponen data blog tersebut dapat ditekan untuk mengalihkan halaman ke

detail blog. Gambar 4.3 menunjukkan tampilan halaman *news*.



Gambar 4.3 Halaman News

4.1.4 Halaman Blog

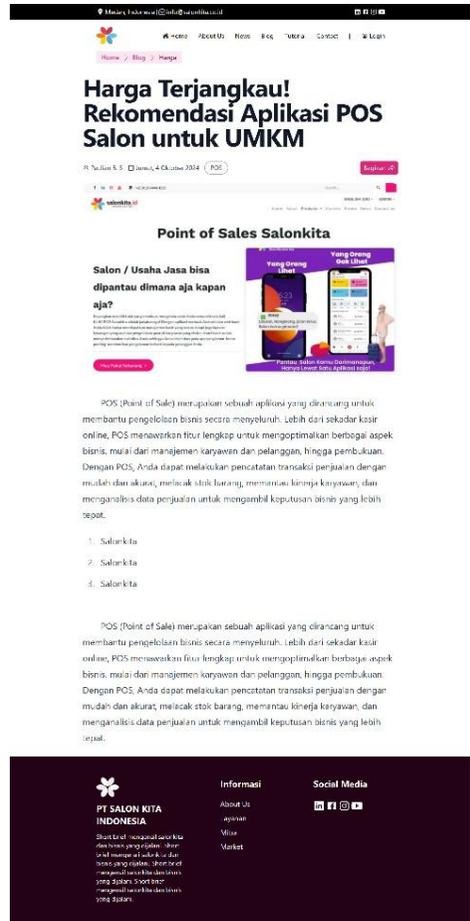


Gambar 4.4 Halaman Blog

Halaman selanjutnya adalah halaman blog yang menjadi pusat pengguna peserta untuk mencari blog bacaan. Data-data blog yang diambil dari server akan dikelompokkan ke dalam segmen-segmen tersendiri berdasarkan kategori dari blog tersebut. Halaman ini bertujuan untuk menampilkan pratinjau dari masing-masing data yang nantinya pengguna bisa baca lebih lanjut dengan menekan komponen blog tersebut. Gambar 4.4 menunjukkan tampilan dari halaman blog.

4.1.5 Halaman Detail Blog

Tampilan yang juga menjadi fitur terakhir *frontend* aplikasi ini adalah detail blog. Pengguna yang mengunjungi halaman utama, *news*, dan blog akan dialihkan ke sini ketika mereka menekan komponen blog yang disajikan kepada mereka. Tidak seperti di halaman lain, halaman ini menampilkan data blog secara penuh. Konten dari blog pada halaman ini juga dilengkapi dengan tampilan *markdown* yang bisa menunjukkan tampilan konten layaknya sebuah dokumen dengan berbagai kustomisasi seperti penomoran data daftar. Gambar 4.5 menggambarkan tampilan halaman program penelitian.



Gambar 4.5 Halaman Detail Blog

V. KESIMPULAN

Berdasarkan berbagai hal yang telah disampaikan pada bab sebelumnya serta perancangan dan implementasi aplikasi yang sudah dilakukan penulis, maka dapat diambil simpulan sebagai berikut :

1. Implementasi aplikasi sudah dapat memberikan UI/UX yang sesuai dengan desain yang didefinisikan pada tahap komunikasi dan perencanaan. Konsep tampilan yang diangkat merupakan konsep aplikasi

modern yaitu *reusable component* yang menjadikan tampilan komponen aplikasi yang sama terlihat konsisten di mana hal ini diadopsi dalam pengembangan web modern.

2. Implementasi telah berhasil menerapkan aplikasi web responsif yang bersifat adaptif pada setiap lebar resolusi layar yang ditunjukkan pada hasil pembahasan bab sebelumnya. Hal ini bermanfaat untuk aksesibilitas dari aplikasi sendiri karena pengguna tidak perlu menggunakan perangkat komputer desktop untuk mengakses aplikasi ini

DAFTAR PUSTAKA

- Pressman, R. S. (2009). *Software Engineering A Practitioner's Approach 7th Ed*
- Roger S. Pressman. In *Software Engineering A Practitioner's Approach 7th Ed*
 - Roger S. Pressman.
 -
- Wasino, Herwindiati, D. E., Setyawan, I. R., & Maupa, H. (2023). *Desain Situs Web Yang Responsif Berdasarkan Strategi Agile Sebagai Pendukung Pemasaran Destinasi Wisata*. *Jurnal Teknik Informatika Dan Sistem Informasi*, 10, 526–540.
- Siahaan, V., & Sianipar, R. H. (2018). *JavaScript: Dari A Sampai Z*. Sparta Publisher
- Bierman, G., Abadi, M., & Torgersen, M. (2014). *Understanding*
- TypeScript. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8586 LNCS.
- Ayuningtyas, D., & Rayhani, M. (2018). *Analisis situasi kesehatan mental pada masyarakat di Indonesia dan strategi penanggulangannya*. *Jurnal Ilmu Kesehatan Masyarakat*, 9(1), 1-10
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language Pengguna Guide (2nd ed.)*. Reading, MA: Addison Wesley
- Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language Pengguna Guide*. Reading, MA: Addison Wesley