



KERJA PRAKTIK – KI141330

**Pembuatan Website Company Profile Perusahaan
Tata Griya Interior**

Aksamedia

Jalan Raya Kedung Cowek No 165, Bulak, Kec. Bulak, Kota
SBY, Jawa Timur 60124, Indonesia

Periode: 1 Februari 2021 - 31 Maret 2021

Oleh:

Pembimbing Jurusan

Imam Kuswardayan, S.Kom, M.T.

Pembimbing Lapangan

Tri Hendry Andhika

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2021



KERJA PRAKTIK – KII141330

**Pembuatan Website Company Profile Perusahaan
Tata Griya Interior**

Aksamedia

Jalan Raya Kedung Cowek No 165, Bulak, Kec. Bulak, Kota
SBY, Jawa Timur 60124, Indonesia

Periode: 1 Februari 2021 - 31 Maret 2021

Oleh:

Pembimbing Jurusan

Imam Kuswardaya, S.Kom, M.T.

Pembimbing Lapangan

Tri Hendry Andhika

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2021

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KERJA PRAKTIK

Pembuatan Website Company Profile Perusahaan Tata Griya Interior

Oleh:

Gema Adi Perwira

05111840000138

Nodas Uziel Putra Serpara

05111840007007

Mengetahui,

Aksamedia

Pembimbing Kerja Praktik



Tri Hendry Andhika

Mengetujui,

Dosen Pembimbing Kerja Praktik



Imam Kuswardhyan, S.Kom, M.T.

NIP.197612151903121001

SURABAYA

11 Juli 2021

[Halaman ini sengaja dikosongkan]

Redesign Website Perusahaan Tata Griya Interior Gresik dan Website Engine Aksamedia

Nama Mahasiswa : Gema Adi Perwira
NRP : 05111840000138
Nama Mahasiswa : Nodas Uziel Putra Serpara
NRP : 05111840007007
Departemen : Informatika FTEIC-ITS
Pembimbing Jurusan : Imam Kuswardayan , S.Kom,
M.T.
Pembimbing Lapangan : Tri Hendry Andhika

ABSTRAK

Dalam meningkatkan daya jual website pada perusahaan Aksamedia, dilakukan desain ulang pada website - website tertentu yang salah satunya adalah Tata Griya Interior. Dan juga diharuskannya perubahan desain Engine Aksamedia agar mengikuti kemajuan pengalaman pengguna dan antarmuka pengguna.

Dengan adanya website Tata Griya Interior yang dibuat dengan menggunakan *framework* laravel ini, akan membantu meningkatkan daya jual website perusahaan Aksamedia ke klien-klien lama. Serta dengan dibuatnya desain yang lebih modern, lalu mengikuti pengalaman pengguna dan antarmuka pengguna yang diperbarui, diharapkan website tersebut dapat memenuhi harapan klien-klien lama yang diperbaharui websitenya.

Kata kunci: Desain, Website

KATA PENGANTAR

Puji syukur kami haturkan kepada Allah SWT karena berkat rahmat-Nya kami dapat melaksanakan salah satu kewajiban kami sebagai mahasiswa Departemen Informatika, yakni Kerja Praktek (KP).

Kami menyadari masih ada kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini. Namun, kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Kami mengharapkan kritik dan saran yang membangun untuk kesempurnaan buku laporan kerja praktik ini.

Melalui buku ini, kami juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu, baik langsung maupun tidak langsung, dalam pelaksanaan kerja praktik hingga penyusunan laporan. Orang-orang tersebut antara lain adalah:

1. Kedua orang tua penulis.
2. Imam Kuswardayam, S.Kom, M.T. selaku dosen pembimbing Kerja Praktik.
3. Mas Tri Hendry Andhika, selaku pembimbing lapangan kami di Aksamedia Surabaya.
4. Teman-teman kami yang satu kos, selaku penyemangat dan membantu pengerjaan kerja praktik kami.

Surabaya, 1 Juli 2021
Gema Adi Perwira, Nodas Uziel Putra Serpara

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

| | |
|--------------------------------|----|
| LEMBAR PENGESAHAN | 4 |
| ABSTRAK | 7 |
| KATA PENGANTAR | 8 |
| DAFTAR ISI | 10 |
| DAFTAR TABEL | 14 |
| DAFTAR GAMBAR | 15 |
| BAB I | |
| PENDAHULUAN | 16 |
| Latar Belakang | 16 |
| Tujuan | 16 |
| Manfaat | 17 |
| Rumusan Masalah | 17 |
| Lokasi dan Waktu Kerja Praktik | 17 |
| Metodologi Kerja Praktik | 17 |
| Sistematika Laporan | 19 |
| BAB II | |
| PROFIL PERUSAHAAN | 22 |
| Profil Aksamedia | 22 |
| Sejarah Singkat Aksamedia | 22 |
| Kegiatan Aksamedia | 22 |
| BAB III | |
| TINJAUAN PUSTAKA | 24 |
| Pemrograman Web | 24 |
| HTML | 24 |

| | |
|--|-----------|
| XAMPP | 25 |
| Bootstrap | 25 |
| Laravel | 25 |
| Visual Studio Code | 26 |
| BAB IV | |
| ANALISIS DAN PERANCANGAN SISTEM | 28 |
| Analisis Sistem | 28 |
| Definisi Umum Website | 28 |
| Analisis Kebutuhan | 29 |
| Kebutuhan Fungsional | 29 |
| Kebutuhan non-Fungsional | 30 |
| Perancangan Sistem | 31 |
| Desain Sistem | 31 |
| Diagram Activity | 31 |
| F01 Melakukan Login | 31 |
| F02 Super admin dapat mengelola website | 32 |
| F03 Super admin dapat membuat, menambahkan dan | 33 |
| menghapus role atau anggota | 33 |
| F04 Admin dapat mengakses admin panel | 34 |
| F05 Writer dapat membuat, menambahkan dan | 35 |
| menghapus postingan blog | 35 |
| F06 Writer dapat membuat, menambahkan dan | 36 |
| menghapus kategori blog | 36 |
| F07 User dapat melihat halaman website | 37 |
| Analisis Website Lama Tata Griya Interior | 37 |
| Halaman Depan | 37 |
| Halaman Portofolio | 38 |

| | |
|--------------------------------|------------|
| Halaman About | 40 |
| Halaman Contact | 40 |
| BAB V | |
| IMPLEMENTASI SISTEM | 43 |
| Implementasi Controller | 43 |
| Controller Authentication | 43 |
| Controller Blog | 46 |
| Controller Permission | 59 |
| Controller Role | 61 |
| Controller User | 71 |
| Implementasi View | 90 |
| View Login | 90 |
| View Dashboard Admin Panel | 90 |
| View Website | 91 |
| BAB VI | |
| PENGUJIAN DAN EVALUASI | 95 |
| Tujuan Pengujian | 95 |
| Kriteria Pengujian | 95 |
| Skenario Pengujian | 95 |
| Evaluasi Pengujian | 96 |
| Evaluasi Oleh Dosen Pembimbing | 99 |
| 6.5.1. SDLC | 99 |
| 6.5.2. Metode Perancangan UX | 102 |
| BAB VII | |
| KESIMPULAN | 105 |
| DAFTAR PUSTAKA | 107 |
| BIODATA PENULIS I | 109 |

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

| | |
|---|----|
| Tabel 4. 1 Karakteristik Pengguna | 31 |
| Tabel 4. 2 Kebutuhan Fungsional | 32 |
| Tabel 4. 3 Kebutuhan Non-Fungsional | 33 |
| | |
| Tabel 6. 1 Evaluasi Pengujian sebagai Admin | 96 |
| Tabel 6. 2 Evaluasi Pengujian sebagai User | 96 |
| Tabel 6. 3 Evaluasi Pengujian sebagai Super Admin | 97 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 4. 1 Diagram Aktivitas Melakukan Login | 34 |
| Gambar 4. 2 Diagram Admin Mengelola Website | 35 |
| Gambar 4. 3 Diagram Aktivitas Admin Membuat, Menambahkan dan Menghapus Role atau Anggota | 36 |
| Gambar 4. 4 Diagram Aktivitas Admin Mengakses Admin Panel | 37 |
| Gambar 4. 5 Diagram Aktivitas Writer Membuat, Menambahkan dan Menghapus Postingan Blog | 38 |
| Gambar 4. 6 Diagram Aktivitas Writer Membuat, Menambahkan dan Menghapus Kategori Blog | 39 |
| Gambar 4.7 Diagram Aktivitas Mengakses Website | 40 |
| | |
| Gambar 5. 1 Login Sistem | 89 |
| Gambar 5. 2 Dashboard Admin Panel | 90 |
| Gambar 5. 3 Halaman Home | 90 |
| Gambar 5. 4 Halaman Tentang Kami | 91 |
| Gambar 5. 5 Halaman Layanan | 91 |
| Gambar 5. 6 Halaman Portofolio | 92 |
| Gambar 5. 7 Halaman Hubungi | 92 |

BAB I

PENDAHULUAN

1.1. Latar Belakang

Semua perusahaan pasti mempunyai cara masing-masing, salah satunya adalah website company profile yang bertujuan untuk memberikan gambaran umum tentang perusahaan. Selain itu, company profile juga untuk memperkuat branding sekaligus alat pemasaran guna menarik banyak client atau customer. Oleh karena itu, website company profile harus berisi tentang detail perusahaan serta pencapaian perusahaan dengan desain yang menarik guna meningkatkan daya tarik.

Pada setiap proyek yang telah dikerjakan oleh perusahaan Aksamedia, dilakukan pengecekan ulang terhadap website-website yang telah ketinggalan antarmuka penggunaannya dan pengalaman penggunaannya, pada saat ditemukan proyek-proyek tersebut, dilakukan desain ulang yang berguna untuk meningkatkan daya jual perusahaan ke klien-klien lama yang tertarik dengan pembaruan website mereka.

Dengan pembaruan yang dilakukan, diharapkan peserta kerja praktik dapat memberikan ide dan kreasi semaksimal mungkin dalam pembaruan website-website yang telah ketinggalan zaman.

1.2. Tujuan

Tujuan kerja praktik ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban dua SKS. Selain itu juga untuk membuat ulang website dari perusahaan Tata Griya Interior dengan desain baru guna memperindah tampilan dan meningkatkan nilai jual.

1.3. Manfaat

Manfaat dari pembuatan website Tata Griya Interior ini adalah sebagai sarana promosi untuk meningkatkan nilai jual dari website yang dibuat oleh perusahaan Aksamedia Surabaya.

1.4. Rumusan Masalah

Berikut ini rumusan masalah pada kerja praktik pembuatan website Tata Griya Interior pada Aksamedia :

1. Bagaimana cara membuat website dengan tampilan yang lebih menarik dari sebelumnya guna meningkatkan nilai jual website itu sendiri?
2. Bagaimana cara membuat website dengan UI/UX yang mudah digunakan oleh setiap orang?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : Kantor Aksamedia dan daring (Hybrid).
Alamat : Jalan Raya Kedung Cowek No 165,
Bulak, Kec. Bulak, Kota SBY, Jawa Timur 60124,
Indonesia / Kos dan rumah masing-masing.
Waktu : 1 Februari 2021 – 31 Maret 2021.
Hari Kerja : Selasa – Kamis
Jam Kerja : 09.00 WIB – 15.00 WIB (Fleksibel)

1.6. Metodologi Kerja Praktik

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

1. Perumusan Masalah

Untuk mengetahui permasalahan apa yang harus diselesaikan, diberikan penjelasan mengenai alasan mengapa diperlukan pembuatan ulang website Tata Griya Interior ini.

Dijelaskan juga mengenai tujuan dari kerja praktik ini. Penjelasan mengenai hal ini dijelaskan oleh pembimbing lapangan kerja praktik. Dari penjelasannya dihasilkan catatan-catatan penting mengenai gambaran desain UI dan UX yang akan dibuat. Dengan begitu dapat diputuskan tools yang digunakan adalah Visual Studio Code dan XAMPP serta menggunakan *framework laravel* dan *framework CSS* yaitu *Bootstrap*.

2. Studi Literatur

Setelah mendapat penjelasan dari pembimbing lapangan, kami diberitahu tinjauan apa saja yang akan diimplementasikan untuk membuat ulang website yang telah ditentukan. Tinjauan yang dipakai meliputi HTML, XAMPP, Laravel, SQL, serta Bootstrap. Pada tahap ini dilakukan proses pencarian, pembelajaran, pengumpulan dan pemahaman informasi serta literatur yang berkaitan untuk membantu dalam implementasi website ini. Informasi bisa didapat dari internet untuk istilah-istilah umum yang digunakan dalam pengimplementasian *framework laravel*.

3. Analisis dan Perancangan

Langkah ini meliputi pengamatan dari website lama dari Tata Griya Interior. Bagaimana UI/UX yang digunakan pada website tersebut. Dari pengamatan tersebut didapatkan kekurangan dari website tersebut. Dari hasil tersebut semakin membuat gambaran bagaimana UI/UX dari website baru yang akan dibuat. Dibuatlah sketsa awal tampilan website yang akan dibuat serta diskusi mengenai sketsa dari tampilan tersebut dan diskusi hal-hal lain yang berkaitan dengan kebutuhan dalam membuat website.

4. Implementasi

Implementasi didasarkan pada perancangan dan analisis sebelumnya. Penentuan desain, animasi, warna, serta fitur yang akan dibuat pada website juga didasari pada analisis sebelumnya sesuai dengan kebutuhan.

Pengerjaan dilakukan dengan pelaporan setiap minggu yang dipantau oleh pembimbing lapangan dengan adanya meeting offline minimal satu kali dalam satu minggu untuk mengetahui progres dan kendala dalam membuat website ini.

5. Pengujian dan Evaluasi

Pengujian dilakukan dengan menguji website yang telah dibuat. Kesesuaian sistem dengan kebutuhan akan menentukan keberhasilan dalam pengujian. Hal ini akan menghasilkan hasil evaluasi apakah sistem sudah sesuai dengan tujuan dan kebutuhan atau belum. Selain itu UI/UX masuk dalam tahap pengujian ini apakah sudah menutupi kekurangan dari website lama atau belum.

6. Kesimpulan dan Saran

Kesimpulan yang kami dapatkan diantaranya adalah perlunya penyesuaian atau pemahaman terhadap masalah yang dibutuhkan untuk membuat suatu website pada tingkat kesulitan yang berbeda-beda serta dibutuhkan komunikasi yang baik dengan pembimbing lapangan agar dapat mengimplementasikan website yang diharapkan dengan lebih baik dari yang sebelumnya.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut:

1. Bab I Pendahuluan

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan, serta sistematika

pengerjaan kerja praktik dan juga penulisan laporan kerja praktik.

2. Bab II Profil Perusahaan

Pada bab ini, dijelaskan secara rinci tentang profil perusahaan tempat kami melaksanakan kerja praktik, yakni Aksamedia.

3. Bab III Tinjauan Pustaka

Pada bab ini, dijelaskan mengenai tinjauan pustaka dan literatur yang digunakan dalam penyelesaian kerja praktik di Aksamedia.

4. Bab IV Analisis dan Perancangan Sistem

Pada bab ini, dijelaskan hasil pembelajaran atau analisis terhadap apa saja yang diperlukan dan harus diperhatikan dalam pengembangan aplikasi yang dikerjakan selama KP.

5. Bab V Implementasi Sistem

Pada bab ini, berisi penjelasan tahap-tahap yang dilakukan untuk proses implementasi pembuatan website dengan tampilan baru.

6. Bab VI Pengujian dan Evaluasi

Pada bab ini, dijelaskan tentang hasil pengujian dan evaluasi dari website yang telah dibuat selama pelaksanaan kerja praktik di Aksamedia.

7. Bab VII Kesimpulan dan Saran

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil Aksamedia

Aksamedia merupakan perusahaan yang bergerak di bidang IT yang membantu client mengubah bisnis mereka secara digital dengan menyediakan layanan pembuatan website dan rekayasa perangkat lunak.

2.2. Sejarah Singkat Aksamedia

Aksamedia berdiri pada tahun 2013. Sejak 2013, Aksamedia telah membantu perusahaan di berbagai industri mengoptimalkan proses bisnis mereka melalui solusi perangkat lunak.

2.3. Kegiatan Aksamedia

Dengan tim yang terkoordinasi dengan baik yang terdiri dari 30+ profesional TI, Kami melayani pelanggan kami di seluruh Indonesia terlepas dari domain operasi, lokasi geografis, dan spesifikasi organisasi mereka. AKSAMEDIA menjawab kebutuhan TI perusahaan dari semua ukuran, mulai dari perusahaan rintisan hingga perusahaan besar, meningkatkan cara mereka bekerja dan meningkatkan nilai yang mereka ciptakan untuk dunia modern.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses kerja praktik.

3.1. Pemrograman Web

Web atau World Wide Web adalah ruang informasi yang berisi dokumen dan resource web lainnya yang dapat diidentifikasi melalui sebuah URL (Uniform Resource Locators, contohnya www.google.com) dan diakses ketika terkoneksi dengan internet. Halaman penyedia dokumen di dalam web dapat disebut sebagai website yang dapat terkoneksi satu dengan lainnya (hyperlink).

Pemrograman web adalah proses pembuatan halaman tersebut agar bisa diakses oleh semua orang. Dalam pembuatan website, diperlukan sebuah standar pada website agar semua orang dapat membaca informasi dalam keadaan yang berbeda. Standar tersebut adalah HTML (Hypertext Markup Language). Jadi pemrograman web memiliki tugas untuk menciptakan suatu halaman sesuai standar HTML agar semua orang memiliki akses pada informasi di dalam halaman tersebut.

3.2. HTML

Bahasa standar internasional yang digunakan untuk membuat halaman web. HTML menggambarkan struktur dan isi semantik dari sebuah dokumen. HTML biasanya digabungkan dengan css dan javascript. css untuk memperindah tampilan dan javascript untuk client side scripting language.

3.3. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (tempat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya.

3.4. Bootstrap

Bootstrap adalah kerangka kerja CSS yang sumber terbuka dan bebas untuk merancang situs web dan aplikasi web. Kerangka kerja ini berisi templat desain berbasis HTML dan CSS untuk tipografi, formulir, tombol, navigasi, dan komponen antarmuka lainnya, serta juga ekstensi opsional JavaScript. Tidak seperti kebanyakan kerangka kerja web lainnya, kerangka kerja ini hanya fokus pada pengembangan front-end saja.

3.5. Laravel

Laravel dikembangkan oleh Taylor Otwell, laravel dibangun mulai april 2011. Laravel adalah framework ekspresif dan sintaks yang elegan. Dibangun lebih menyenangkan yang sudah melalui pengalaman yang kreatif agar membuat produk jadi memuaskan. Laravel mencoba mengurangi proses routing, caching dan session sehingga membantu proses develop lebih menyenangkan tanpa merubah fungsionalitas. Aplikasi ini menggunakan laravel karena laravel merupakan framework terbaik tahun 2015.

Survey dari web Sitepoint.com pada Maret 2015 framework PHP paling populer, Laravel menempati posisi teratas. Karena menjadi aplikasi terbaik dan paling populer, pengguna laravel juga lebih banyak sehingga memudahkan untuk mencari solusi permasalahan dalam pembuatan aplikasi. Laravel juga menyediakan dokumentasi yang lengkap. Selain kemudahan pembangunan aplikasi menggunakan laravel, aplikasi berbasis web tidak harus dibatasi dengan platform ataupun sistem operasi karena pengguna hanya perlu mengakses aplikasi menggunakan web browser, dengan syarat pengguna masih terkoneksi internet. Karena aplikasi ditujukan untuk multi platform, maka aplikasi haruslah bisa menyesuaikan hasil tampilan dengan perangkat pengguna

3.6. Visual Studio Code

Visual Studio Code adalah editor source code yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi.

Visual Studio Code gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi proprietary. Kode Visual Studio didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk desktop yang berjalan pada Blink layout. Meskipun menggunakan kerangka Elektron, Visual Studio Code tidak menggunakan Atom dan menggunakan komponen editor yang sama (diberi kode nama "Monaco") yang digunakan dalam Visual Studio Team Services yang sebelumnya disebut Visual Studio Online (Lardinois, 2015).

[Halaman ini sengaja dikosongkan]

BAB IV ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Sistem

Pada bab ini akan dijelaskan mengenai tahapan dalam pembuatan website Tata Griya Interior. Hal tersebut dijelaskan ke dalam dua bagian, definisi umum website Tata Griya Interior dan analisis kebutuhan.

4.1.1. Definisi Umum Website

Secara umum, website Tata Griya Interior adalah website company profile dari perusahaan Tata Griya Interior. Perusahaan ini bergerak dibidang pengerjaan desain interior seperti kitchen set, bedroom, living room, front office, dll. Website ini memiliki empat role yakni dari pihak super admin, admin, writer dan user.

Tabel 4. 1 Karakteristik Pengguna

| Kategori Role | Tugas | Hak Akses ke Website |
|----------------------|---|---------------------------------------|
| Super Admin | Mengawasi website Mengelola website Membuat role baru Menambahkan anggota dari masing-masing role Menghapus role maupun anggota | Semua Fitur |
| Admin | Mengakses admin panel Membantu tugas dari super admin | Akses admin panel dan halaman website |

| | | |
|--------|---|---------------------------------------|
| Writer | Membuat postingan blog Mengedit postingan blog Menghapus postingan blog Membuat kategori blog Mengedit kategori blog Menghapus kategori blog | Akses admin panel dan halaman website |
| User | Hanya dapat melihat tampilan website | Halaman website |

4.1.2. Analisis Kebutuhan

Dalam website ini, terdapat fungsi-fungsi yang harus dipenuhi oleh sistem. Kebutuhan ini terbagi ke dalam dua jenis, yakni kebutuhan fungsional dan kebutuhan non-fungsional.

4.1.2.1. Kebutuhan Fungsional

Kebutuhan fungsional pada website ini menjelaskan bagaimana sistem ini bekerja yang akan dijelaskan pada Tabel 4.2.

Tabel 4. 2 Kebutuhan Fungsional

| Kode Kebutuhan | Deskripsi Kebutuhan | Modul |
|----------------|--|--|
| F01 | Melakukan Login | Modul Login |
| F02 | Super admin dapat mengelola website | Modul admin panel |
| F03 | Super admin dapat membuat, menambahkan, dan menghapus role atau anggota. | Modul admin panel bagian role & permission |
| F04 | Admin dapat mengakses admin panel | Modul admin panel |

| | | |
|-----|--|-----------------------------------|
| F05 | Writer dapat membuat, mengedit, dan menghapus postingan blog | Modul admin panel bagian blog |
| F06 | Writer dapat membuat, mengedit, dan menghapus kategori blog | Modul admin panel bagian category |
| F07 | User dapat melihat halaman website | Halaman website |

4.1.2.2. Kebutuhan non-Fungsional

Tabel 4. 3 Kebutuhan Non-Fungsional

| Kode Kebutuhan | Deskripsi Kebutuhan | Kualitas |
|----------------|---|-------------|
| NF01 | Hanya akun pengguna yang teridentifikasi dan terotorisasi yang dapat membuka dan mengakses admin panel. | Security |
| NF02 | Sistem dapat diakses 24 jam | Portability |
| NF03 | Sistem dibuat dalam bentuk website | Portability |
| NF04 | Sistem dapat diakses pada semua <i>web-browser</i> | Portability |

4.2. Perancangan Sistem

4.2.1. Desain Sistem

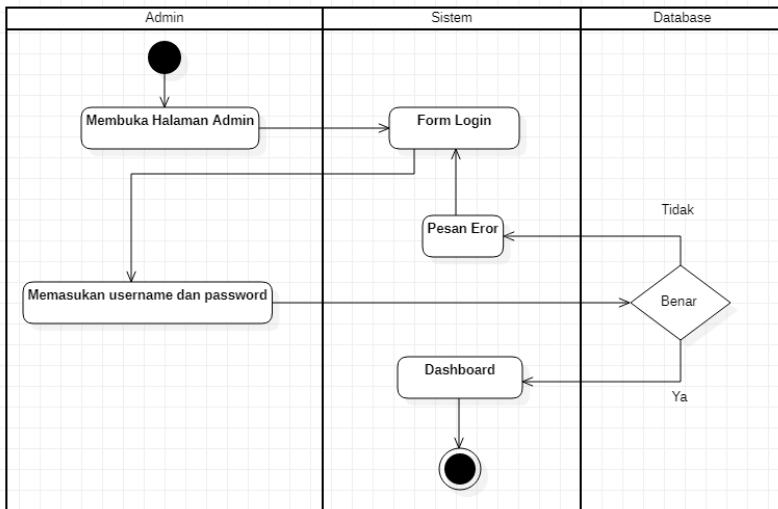
Desain sistem digunakan untuk mengetahui definisi dan rincian tiap kebutuhan fungsional serta persiapan untuk membangun website. Desain sistem yang digunakan adalah *Activity Diagram*.

Activity Diagram menunjukkan bagaimana sistem berinteraksi dengan *stakeholder*.

4.2.1.1. Diagram Activity

F01 Melakukan Login

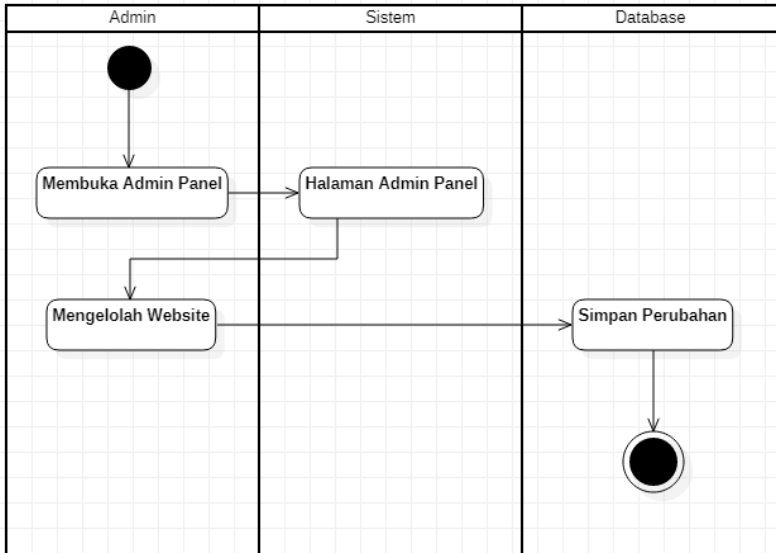
Stakeholder melakukan *login* kedalam aplikasi untuk memastikan keaslian hak akses. Gambar 4.4. di bawah ini merupakan diagram aktivitas yang menunjukkan alur autentikasi



Gambar 4. 1 Diagram Aktivitas Melakukan Login

F02 Super admin dapat mengelola website

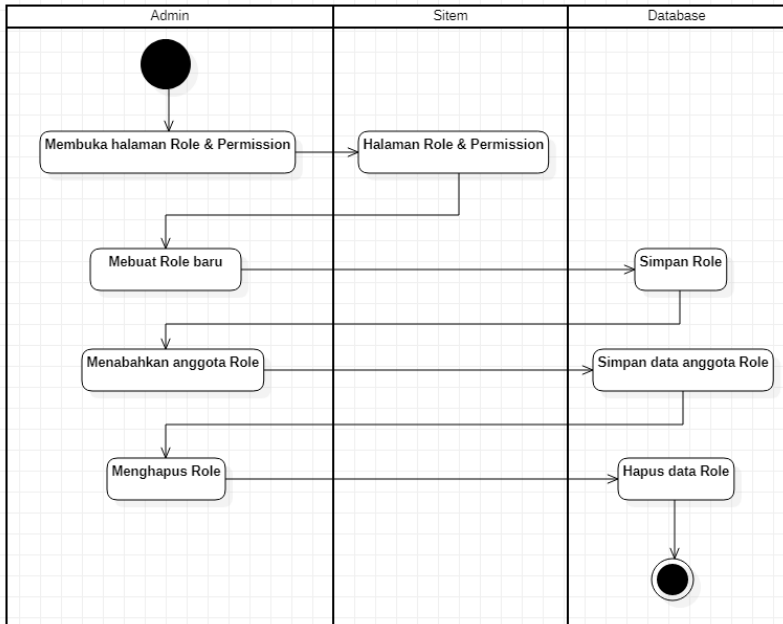
Super Admin dapat melakukan pengelolaan website sesuai diagram aktivitas sebagai berikut :



Gambar 4.2 Diagram Aktivitas Admin Mengelola Website

F03 Super admin dapat membuat, menambahkan dan menghapus role atau anggota

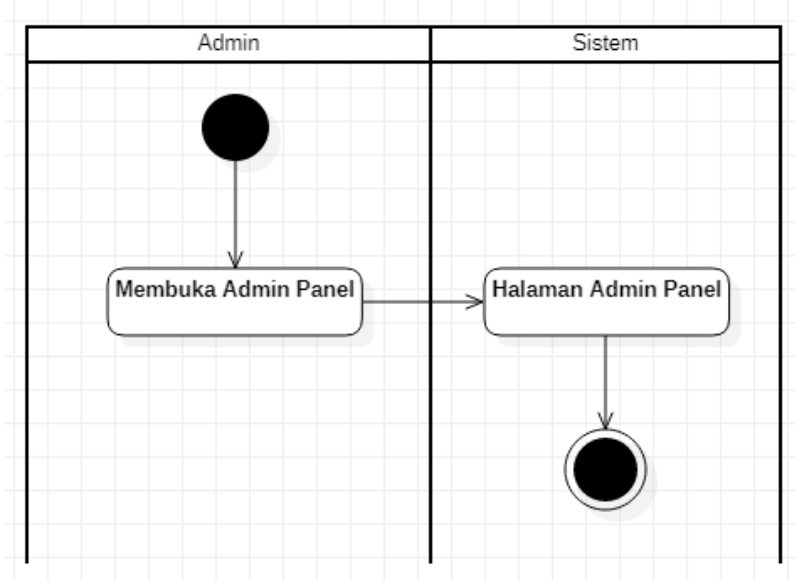
Admin dapat membuat, menambahkan, dan menghapus role atau anggota yang dapat mengakses halaman admin panel.



Gambar 4. 3 Diagram Aktivitas Admin Membuat, Menambahkan dan Menghapus Role atau Anggota

F04 Admin dapat mengakses admin panel

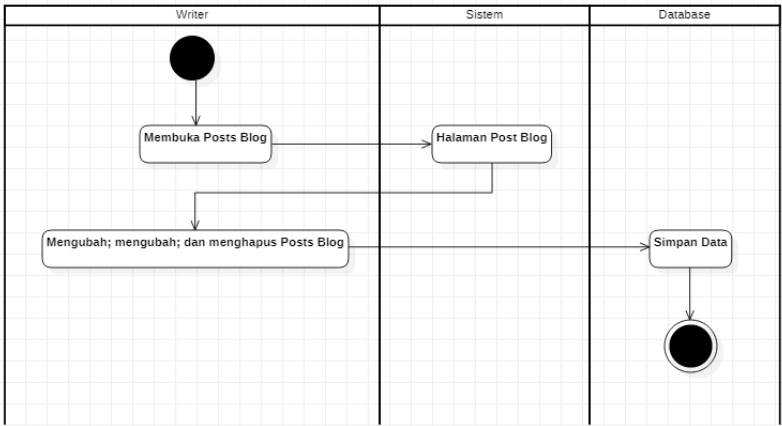
Admin dapat mengakses halaman admin panel.



Gambar 4. 4 Diagram Aktivitas Admin Mengakses Admin Panel

F05 Writer dapat membuat, menambahkan dan menghapus postingan blog

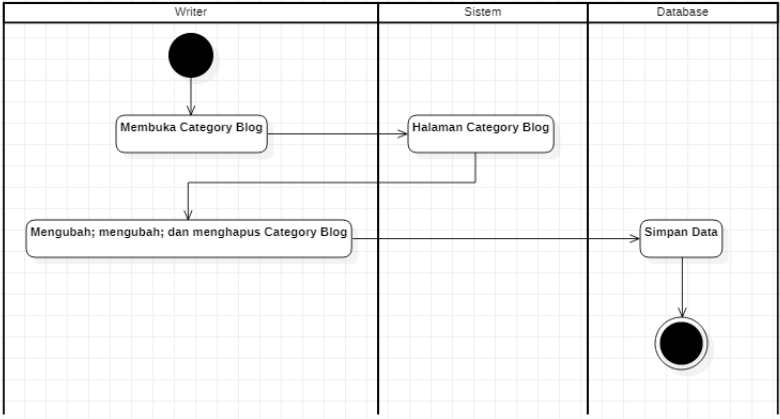
Writer dapat membuat, menambahkan, dan menghapus postingan blog pada website melalui halaman posts blog.



Gambar 4. 5 Diagram Aktivitas Writer Membuat, Menambahkan dan Menghapus Postingan Blog

F06 Writer dapat membuat, menambahkan dan menghapus kategori blog

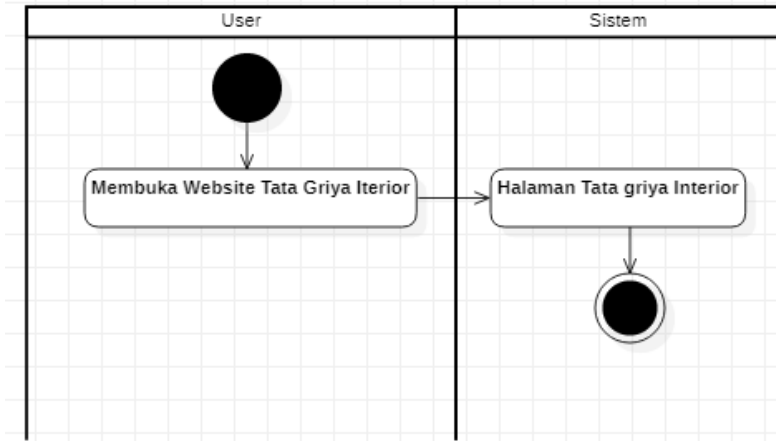
Writer dapat membuat, menambahkan, dan menghapus kategori blog pada website melalui halaman category blog.



Gambar 4. 6 Diagram Aktivitas Writer Membuat, Menambahkan dan Menghapus Kategori Blog

F07 User dapat melihat halaman website

User dapat mengakses website Tata Griya Interior.



Gambar 4. 7 Diagram Aktivitas Mengakses Website

4.3. Analisis Website Lama Tata Griya Interior

4.3.1. Halaman Depan



Pada bagian halaman depan ini terlihat bahwa desain dan struktur konten yang perlu diperbaiki.

Warna yang dipakai pada website ini cenderung gelap dan desainnya yang terlalu ramai.

4.3.2. Halaman Portofolio

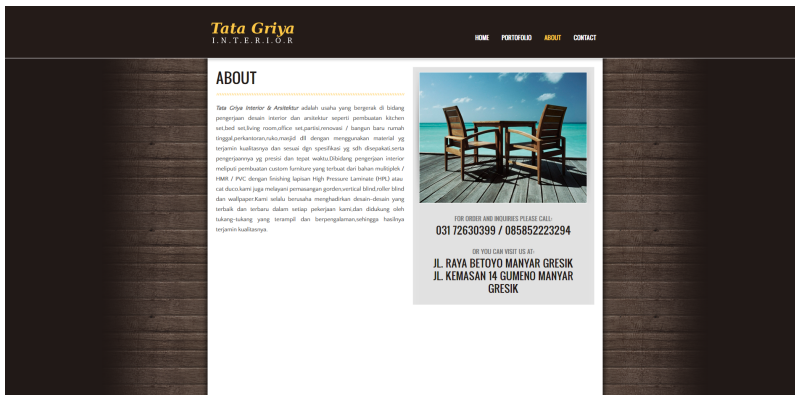


Masalah pada bagian halaman portofolio kurang lebih sama seperti halaman home, yang membedakan disini adalah banyaknya ruang kosong di bagian bawah tulisan. Hal tersebut tentunya sangat tidak efektif dan mengurangi estetika.



Navbar menu portofolio ketika dihover juga akan otomatis muncul menu dropdown seperti gambar diatas, tetapi justru hal itu menambah kekurangan dari website ini. Menu drop down tersebut berwarna sama semua dan tidak ada batas antara portofolio satu dengan portofolio yang lainnya.

4.3.3. Halaman About



Pada halaman about, masalahnya juga pada banyaknya ruang kosong di bagian bawah tulisan. Gambar yang ada pada bagian sisi kanan halaman juga tidak relate dengan tema dari website ini yang bertemakan desain interior. Perbandingan font dari tulisan nomor hp & alamat dengan tulisan informasi perusahaan di sebelah kiri sangatlah kontras, karena ukuran font yang berbeda jauh.

4.3.4. Halaman Contact



Pada halama contact, masalahnya juga pada perbandingan ukuran tulisan antara tulisan di bagian kiri halaman dan bagian kanan halaman. Disini juga sebaiknya gambar yang ada diganti dengan peta lokasi dari perusahaan itu berada.

[Halaman ini sengaja dikosongkan]

BAB V IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari sistem yang kami buat. Implementasi ini kami menggunakan *query native* yang dibagi menjadi beberapa bagian, yakni implementasi lapisan kontrol, dan implementasi *view* atau antarmuka pengguna.

5.1. Implementasi Controller

Implementasi lapisan kontrol ini berisi logika yang digunakan aplikasi seperti kontrol untuk memasukkan data ke *database*, menampilkan data, dan logika lainnya. Di dalam pengembangannya, kami memilih untuk memisahkan *Controller* memudahkan *maintenance* dengan mempertahankan *readability*.

5.2.1. Controller Authentication

Controller ini berguna untuk mengatur akses login dan logout pada halaman admin panel. Berikut adalah potongan code pada LoginController dan LogoutController :

```
<?php

namespace App\Http\Controllers\Admin\Auth;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\User;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    public function index(){
```

```

        return view('admin.auth.login');
    }

    public function store(Request $request){
        $this->validate($request,[
            'username' => 'required',
            'password' => 'required'
        ]);

        $credentials = [
            'username' => $request->username,
            'password' => $request->password
        ];
        $remember = $request->remember;

        $admin =
        auth()->guard('web')->attempt($credentials,
        $remember);

        if($admin){
            return
            redirect()->intended(route('admin.index'));
        }

        return
        redirect()->back()->withInput($request->only('user
        name','remember'));
    }

    public function logout()

```



```
    {
        Auth::logout();

        return redirect()->back();
    }
}
```

```
<?php

namespace
App\Http\Controllers\Admin\Auth;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class LogoutController extends Controller
{
    public function index(){
        auth()->guard('web')->logout();
        return
        redirect()->route('admin.auth.login');
    }
}
```

5.2.2. Controller Blog

Lapisan kontrol ini berguna untuk mengatur kategori dan postingan dari blog pada sebuah website. Berikut adalah potongan code pada *CategoryController* :

```
1. <?php
2.
3. namespace App\Http\Controllers\Admin\Blog;
4.
5. use Illuminate\Http\Request;
6. use App\Http\Controllers\Controller;
7. use App\Models\Blog\Category;
8.
9. class CategoryController extends Controller
10. {
11.
12.     public function __construct() {
13.         $this->middleware('permission:blog categories
14.         read');
15.
16.         $this->middleware('permission:blog categories
17.         create')->only('create', 'store');
```

```

15.         $this->middleware('permission:blog categories
edit')->only('edit', 'update');

16.         $this->middleware('permission:blog categories
delete')->only('destroy');

17.

18.         view()->share('menuActive', 'blog');

19.         view()->share('subMenuActive', 'blog-categories');

20.     }

21.

22.     /**

23.      * Display a listing of the resource.

24.      *

25.      * @return \Illuminate\Http\Response

26.      */

27.     public function index()

28.     {

29.         $data['models'] = Category::orderBy('id',
'desc')->paginate(10);

30.

31.         return view('admin.blog.categories.index', $data);

```

```

32.     }
33.
34.     /**
35.      * Show the form for creating a new resource.
36.      *
37.      * @return \Illuminate\Http\Response
38.      */
39.     public function create()
40.     {
41.         return view('admin.blog.categories.create');
42.     }
43.
44.     /**
45.      * Store a newly created resource in storage.
46.      *
47.      * @param \Illuminate\Http\Request $request
48.      * @return \Illuminate\Http\Response
49.      */

```

```
50.     public function store(Request $request)
51.     {
52.         $request->validate([
53.             'name' => 'required|max:200',
54.             'description' => 'required',
55.         ]);
56.
57.         $category = new Category($request->all());
58.         $category->save();
59.
60.         return redirect()
61.             ->route('admin.blog.categories.index')
62.             ->with([
63.                 'message' => 'Save Successfully'
64.             ]);
65.     }
66.
67.     /**
```

```

68.     * Display the specified resource.
69.
70.     * @param int $id
71.     * @return \Illuminate\Http\Response
72.     */
73. public function show($id)
74. {
75.     //
76. }
77.
78. /**
79.  * Show the form for editing the specified resource.
80.  *
81.  * @param Category $category
82.  * @return \Illuminate\Http\Response
83.  */
84. public function edit(Category $category)
85. {

```

```

86.         return view('admin.blog.categories.edit', ['model'
=> $category]);
87.     }
88.
89.     /**
90.      * Update the specified resource in storage.
91.      *
92.      * @param \Illuminate\Http\Request $request
93.      * @param Category $category
94.      * @return \Illuminate\Http\Response
95.      */
96.     public function update(Request $request, Category
$category)
97.     {
98.         $request->validate([
99.             'name' => 'required|max:200',
100.            'description' => 'required',
101.        ]);
102.

```

```

103.         if ($category->update($request->all())) {
104.             return
105.             redirect()->route('admin.blog.categories.index')->with(['sta
106.             tus' => 'success', 'message' => 'Update Successfully']);
107.         }
108.
109.         return
110.         redirect()->route('admin.blog.categories.edit',
111.         $category->id)->with(['status' => 'danger', 'message' =>
112.         'Update Failed, Contact Developer']);
113.     }
114.
115.     /**
116.     * Remove the specified resource from storage.
117.     *
118.     * @param Category $category
119.     * @return \Illuminate\Http\Response
120.     * @throws \Exception
121.     */
122.     public function destroy(Category $category)
123.     {

```



```

119.         if ($category->delete()) {
120.             return
121.             redirect()->route('admin.blog.categories.index')->with(['sta
122.             tus' => 'success', 'message' => 'Delete Successfully']);
123.         }
124.         return
125.         redirect()->route('admin.blog.categories.index')->with(['sta
126.         tus' => 'danger', 'message' => 'Delete Failed, Contact
127.         Developer']);
128.     }
129. }

```

Kemudian berikut ini adalah potongan code pada PostController :

```

<?php

namespace App\Http\Controllers\Admin\Blog;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Models\Blog\Category;
use App\Models\Blog\Post;
use Illuminate\Support\Facades\Storage;

class PostController extends Controller
{

    public function __construct()

```

```

{
    $this->middleware('permission:blog posts read');
        $this->middleware('permission:blog posts
create')->only('create', 'store');
        $this->middleware('permission:blog posts
edit')->only('edit', 'update');
        $this->middleware('permission:blog posts
delete')->only('destroy');

    view()->share('menuActive', 'blog');
    view()->share('subMenuActive', 'blog-posts');
}

/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
    $data['models'] = Post::orderBy('id',
'desc')->paginate(10);
    return view('admin.blog.posts.index', $data);
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */

```

```

public function create()
{
    $data['categories'] = Category::all();
    return view('admin.blog.posts.create', $data);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'title' => 'required|max:200',
        'description' => 'required',
        'image' => 'required|image',
        'published_at' => ['required', 'date']
    ]);

    $post = new Post($request->all());

    $image = $post->uploadImage($request->file('image'),
'ugc/blog');
    $post->image = $image->lg;

    $post->save();

    return redirect()

```

```

        ->route('admin.blog.posts.index')
            ->with(['status' => 'success', 'message' =>
'Save Successfully']);
    }

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param Post $post
 * @return \Illuminate\Http\Response
 */
public function edit(Post $post)
{
    $data['categories'] = Category::all();
    $data['model'] = $post;
    return view('admin.blog.posts.edit', $data);
}

/**

```

```

* Update the specified resource in storage.
*
* @param \Illuminate\Http\Request $request
* @param Post $post
* @return \Illuminate\Http\Response
*/
public function update(Request $request, Post $post)
{
    $request->validate([
        'title' => 'required|max:200',
        'description' => 'required',
        'image' => 'image',
        'published_at' => ['required', 'date']
    ]);

    if ($request->hasFile('image')) {
        $newImage =
$post->uploadImage($request->file('image'), 'ugc/blog');

        $payload = $request->all();

        if ($newImage) {
            $payload['image'] = $newImage->lg;

            $post->deleteImage();
        }

        $post->update($payload);
    } else {
        $post->update($request->all());
    }
}

```

```

    }

    return
    redirect()->route('admin.blog.posts.index')->with(['status' => 'success', 'message' => 'Update Successfully']);
}

/**
 * Remove the specified resource from storage.
 *
 * @param Post $post
 * @return \Illuminate\Http\Response
 * @throws \Exception
 */
public function destroy(Post $post)
{
    $image = new Post();

    if (Storage::exists($post->image->path)) {
        $image->deleteImage($post->image->path);
    }

    if ($post->delete()) {
        return
        redirect()->route('admin.blog.posts.index')->with(['status' => 'success', 'message' => 'Delete Successfully']);
    }

    return
    redirect()->route('admin.blog.posts.index')->with(['status' => 'success', 'message' => 'Delete Successfully']);
}

```

```

s' => 'danger', 'message' => 'Delete Failed, Contact
Developer']);
    }
}

```

5.2.3. Controller Permission

```

<?php

namespace App\Http\Controllers\Admin\User;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Spatie\Permission\Models\Permission;
use Spatie\Permission\Models\Role;

class PermissionController extends Controller
{

    public function __construct() {
        $this->middleware('role:superadmin');
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
}

```

```

public function index(Role $role)
{
    $role->load('permissions');
    $allPermissions = Permission::get();

    return
view('admin.users.roles.permissions.index', compact(
    'role',
    'allPermissions'
));
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request, Role
$role)
{
    $permissionNames =
collect($request->except('_token'))
    ->keys()
    ->map(function ($key) {
        return str_replace('_', ' ', $key);
    });
}

```



```
        $permissions = Permission::whereIn('name',
$permissionNames)->get();

        $role->syncPermissions($permissions);

        return redirect()->back()->with([
            'status' => 'success',
            'message' => 'Permissions are updated.'
        ]);
    }
}
```

5.2.4. Controller Role

```
1. <?php
2.
3. namespace App\Http\Controllers\Admin\User;
4.
5. use App\Http\Controllers\Controller;
6. use Illuminate\Http\Request;
```

```
7. use Illuminate\Validation\Rule;

8. use Spatie\Permission\Models\Role;

9.

10. class RoleController extends Controller

11. {

12.

13.     public function __construct() {

14.         $this->middleware('permission:roles

15.         read');

16.         $this->middleware('permission:roles

17.         create')->only('create', 'store');

18.         $this->middleware('permission:roles

19.         update')->only('edit', 'update');

20.         $this->middleware('permission:roles

21.         delete')->only('destroy');

22.

23.         view()->share('menuActive', 'users');
```

```
20.         view()->share('subMenuActive',
    'roleAndPermissions');
21.     }
22.
23.     /**
24.      * Display a listing of the resource.
25.      *
26.      * @return \Illuminate\Http\Response
27.      */
28.     public function index(Request $request)
29.     {
30.         $roles = Role::withCount('permissions',
    'users')
31.             ->paginate($request->take ?? 12);
32.
```

```
33.         return view('admin.users.roles.index',
compact('roles'));

34.     }

35.

36.     /**

37.      * Show the form for creating a new
resource.

38.      *

39.      * @return \Illuminate\Http\Response

40.      */

41.     public function create()

42.     {

43.         return view('admin.users.roles.create');

44.     }

45.

46.     /**
```

```
47.     * Store a newly created resource in
      storage.
48.     *
49.     * @param  \Illuminate\Http\Request
      $request
50.     * @return \Illuminate\Http\Response
51.     */
52.     public function store(Request $request)
53.     {
54.         $request->validate([
55.             'name' => [
56.                 'required',
57.                 Rule::unique('roles')
58.             ],
59.         ]);
60.
```

```

61.         Role::create($request->all());
62.
63.         return
        redirect()->route('admin.roles.index')->with([
64.             'status' => 'success',
65.             'message' => 'New role has been
        created.'
66.         ]);
67.     }
68.
69.     /**
70.      * Display the specified resource.
71.      *
72.      * @param int $id
73.      * @return \Illuminate\Http\Response
74.      */

```

```
75.     public function show($id)
76.     {
77.         //
78.     }
79.
80.     /**
81.      * Show the form for editing the specified
      resource.
82.      *
83.      * @param Role $role
84.      * @return \Illuminate\Http\Response
85.      */
86.     public function edit(Role $role)
87.     {
88.         return view('admin.users.roles.edit',
            compact('role'));

```

```

89.     }
90.
91.     /**
92.      * Update the specified resource in storage.
93.      *
94.      * @param \Illuminate\Http\Request
    $request
95.      * @param Role $role
96.      * @return \Illuminate\Http\Response
97.      */
98.     public function update(Request $request,
    Role $role)
99.     {
100.         $request->validate([
101.             'name' => [
102.                 'required',

```



```
103.     Rule::unique('roles')->ignore($role)
104.         ],
105.     ]);
106.
107.     $role->update($request->all());
108.
109.     return redirect()->back()->with([
110.         'status' => 'success',
111.         'message' => 'Role has been
112.         updated.'
113.     ]);
114.
115.     /**
```

```
116.     * Remove the specified resource from
      storage.
117.     *
118.     * @param Role $role
119.     * @return \Illuminate\Http\Response
120.     */
121.     public function destroy(Role $role)
122.     {
123.         if ($role->name === 'superadmin') {
124.             return redirect()->back()->with([
125.                 'status' => 'warning',
126.                 'message' => 'Role superadmin
      can\'t be delete'
127.             ]);
128.         }
129.
```

```

130.     $role->delete();
131.
132.     return redirect()->back()->with([
133.         'status' => 'success',
134.         'message' => 'Role has been
        deleted.'
135.     ]);
136. }
137. }

```

5.2.5. Controller User

Lapisan kontrol ini berguna untuk mengatur login user yang dihubungkan pada SSO. Berikut adalah potongan code pada *User Controller* :

```

138. <?php
139.
140. namespace App\Http\Controllers\Admin\User;
141.

```

```
142. use App\Http\Controllers\Controller;

143. use App\User;

144. use Illuminate\Http\Request;

145. use Illuminate\Support\Facades\Hash;

146. use Illuminate\Validation\Rule;

147. use Spatie\Permission\Models\Role;

148.

149. class UserController extends Controller

150. {

151.

152.     public function __construct() {

153.         $this->middleware('permission:users
read');

154.         $this->middleware('permission:users
create')->only('create', 'store');
```

```
155.     $this->middleware('permission:users
edit')->only('edit', 'update');

156.     $this->middleware('permission:users
delete')->only('destroy');

157.

158.     view()->share('menuActive', 'users');

159.     view()->share('subMenuActive', 'users');

160. }

161.

162. /**
163.  * Display a listing of the resource.
164.  *
165.  * @return \Illuminate\Http\Response
166.  */
167. public function index(Request $request)
168. {
```

```
169.     $users = User::with('roles')
170.         ->select('users.*')
171.         ->leftJoin('model_has_roles',
function ($join) {
172.     $join->on('model_has_roles.model_id', '=',
'users.id')
173.     ->where('model_has_roles.model_type', '=',
User::class);
174.     })
175.     ->leftJoin('roles', 'roles.id', '=',
'model_has_roles.role_id')
176.     ->orderBy('roles.id')
177.     ->orderBy('users.name')
178.     ->paginate($request->take ?: 10);
179.
```

```
180.         return view('admin.users.index',
compact('users'));

181.     }

182.

183.     /**

184.      * Show the form for creating a new
resource.

185.      *

186.      * @return \Illuminate\Http\Response

187.      */

188.     public function create()

189.     {

190.         $admin = auth()->user();

191.         $roles = Role::get();

192.

193.         // return $user;
```

```

194.
195.         return view('admin.users.create',
compact('admin', 'roles'));
196.     }
197.
198.     /**
199.      * Store a newly created resource in
storage.
200.      *
201.      * @param  \Illuminate\Http\Request
$request
202.      * @return \Illuminate\Http\Response
203.      */
204.     public function store(Request $request)
205.     {
206.         $request->validate([

```



```
207.         'name' => ['required'],
208.
209.         'username' => [
210.             'required',
211.             Rule::unique('users')
212.         ],
213.
214.         'email' => [
215.             'required',
216.             Rule::unique('users')
217.         ],
218.
219.         'password' => [
220.             'required',
221.             'confirmed'
```

```
222.         Rule::notIn(['superadmin'])
223.     ],
224.     'image' => ['image']
225.   ]);
226.
227.   $user = new User();
228.   $user->password =
Hash::make($request->password);
229.
230.   if ($request->file('image')) {
231.       $newImage =
$user->uploadImage($request->file('image'),
'ugc/images');
232.   }
233.
234.   $user->name = $request->name;
```

```
235.     $user->username = $request->username;

236.     $user->email = $request->email;

237.

238.     if (@$newImage) {

239.         $user->image = @$newImage->lg;

240.     }

241.

242.     $user->save();

243.

244.     $user->syncRoles(Role::whereName($request->role)
->firstOrFail());

245.

246.     return
    redirect()->route('admin.users.edit',
    [$user->id])->with([

247.         'status' => 'success',
```

```
248.         'message' => 'User has been created'
249.     });
250. }
251.
252. /**
253.  * Display the specified resource.
254.  *
255.  * @param int $id
256.  * @return \Illuminate\Http\Response
257.  */
258. public function show($id)
259. {
260.     //
261. }
262.
```

```
263.     /**
264.      * Show the form for editing the specified
265.      * resource.
266.      *
267.      * @param User $user
268.      * @return \Illuminate\Http\Response
269.      */
270.     public function edit(User $user)
271.     {
272.         $user->load('roles');
273.
274.         $roles = Role::get();
275.
276.         return view('admin.users.edit',
compact('user', 'roles'));
    }
```

```
277.  
278.     /**  
279.      * Update the specified resource in storage.  
280.      *  
281.      * @param \Illuminate\Http\Request  
      $request  
282.      * @param User $user  
283.      * @return \Illuminate\Http\Response  
284.      */  
285.     public function update(Request $request,  
      User $user)  
286.     {  
287.         $request->validate([  
288.             'name' => ['required'],  
289.             'username' => [  
290.                 'required',
```

```
291.     Rule::unique('users')->ignore($user)
292.         ],
293.         'email' => [
294.             'required',
295.             Rule::unique('users')->ignore($user)
296.         ],
297.         'password' => [
298.             Rule::requiredIf($request->old_password),
299.             'confirmed'
300.         ],
301.         'password_confirmation' =>
302.             [Rule::requiredIf($request->password)],
303.         'role' => [
304.             Rule::notIn(['superadmin'])
```

```
304.         ],
305.         'image' => ['image']
306.     ]);
307.
308.     if ($request->old_password) {
309.         if
310.         (!Hash::check($request->old_password,
311.         $user->password)) {
312.             return redirect()
313.                 ->back()
314.                 ->withErrors([
315.                     'old_password' => 'Old
316.                     password is incorrect'
317.                 ])
318.                 ->withInput();
319.         }
320.     }
321. }
```



```
317.  
  
318.         $user->password =  
           Hash::make($request->password);  
  
319.     }  
  
320.  
  
321.         $admin = auth()->user();  
  
322.  
  
323.         if ($request->role &&  
           $admin->hasRole('superadmin') &&  
           !$user->hasRole('superadmin')) {  
  
324.             # ganti role nya  
  
325.             $user->syncRoles(Role::whereName($request->role)  
               ->firstOrFail());  
  
326.         }  
  
327.  
  
328.         if ($request->file('image')) {
```

```
329.         $newImage =
330.         $user->uploadImage($request->file('image'),
331.         'ugc/images');
332.
333.     }
334.
335.     # hapus image kalau sebelumnya udah ada
336.
337.     if (@$newImage && $user->image) {
338.
339.         $user->deleteImage($user->image->path);
340.
341.     }
342.
343.     $user->name = $request->name;
344.
345.     $user->username = $request->username;
346.
347.     $user->email = $request->email;
348.
349.
350.     if (@$newImage) {
```

```
342.         $user->image = @$newImage->lg;
343.     }
344.
345.     $user->save();
346.
347.     return redirect()->back()->with([
348.         'status' => 'success',
349.         'message' => 'User has been updated'
350.     ]);
351. }
352.
353. /**
354.  * Remove the specified resource from
355.  * storage.
```

```

356.     * @param User $user
357.
358.     * @return \Illuminate\Http\Response
359.
360.     */
361.     public function destroy(User $user)
362.     {
363.
364.         $role = $user->roles()->first();
365.
366.         $admin = auth()->user();
367.
368.         if ($role && $role->name ===
369.             'superadmin') {
370.
371.             return redirect()->back()->with([
372.
373.                 'status' => 'warning',
374.
375.                 'message' => 'Superadmin can\'t
376.                 be delete.'
377.
378.             ]);
379.
380.         }

```

```
370.  
371.     if ($admin->id === $user->id) {  
372.         return redirect()->back()->with([  
373.             'status' => 'warning',  
374.             'message' => 'You can\'t delete  
your self.'  
375.         ]);  
376.     }  
377.  
378.     $user->delete();  
379.  
380.     return redirect()->back()->with([  
381.         'status' => 'info',  
382.         'message' => 'User deleted.'  
383.     ]);
```

```
384. }
```

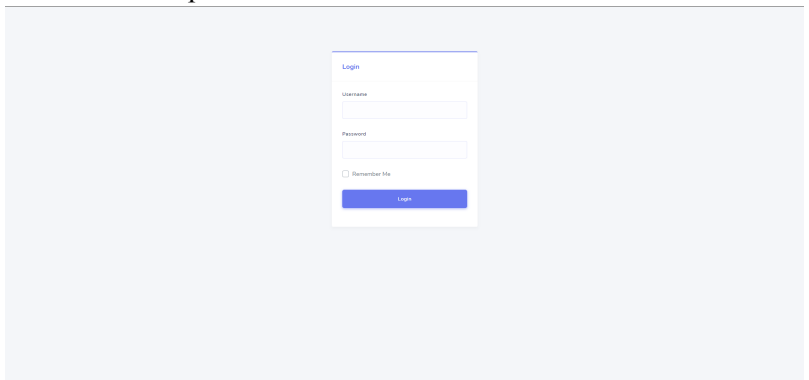
```
385. }
```

5.2. Implementasi View

Pada bagian ini kami akan menampilkan antarmuka halaman yang ada pada website Tata Griya Interior dan halaman admin panel. Berikut tampilan antarmuka website Tata Griya Interior dan halaman admin panel

5.3.1.View Login

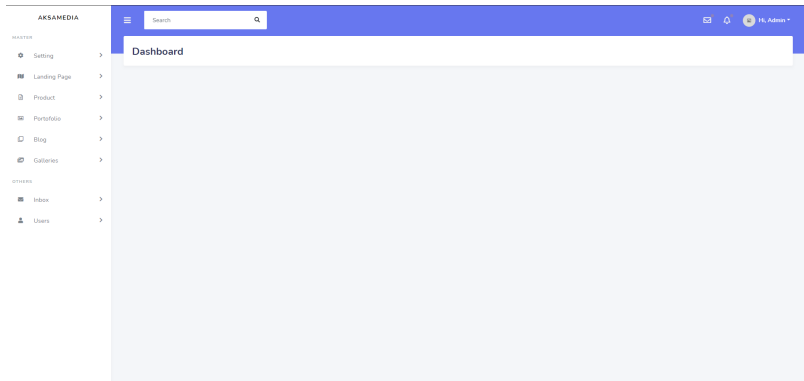
Gambar 5.1 berisi antarmuka login untuk masuk ke halaman admin panel.



Gambar 5.1 Login Sistem

5.3.2.View Dashboard Admin Panel

Gambar 5.2 berisi antarmuka halaman awal admin panel bagi pengguna yang memiliki hak akses sebagai super admin ataupun admin.

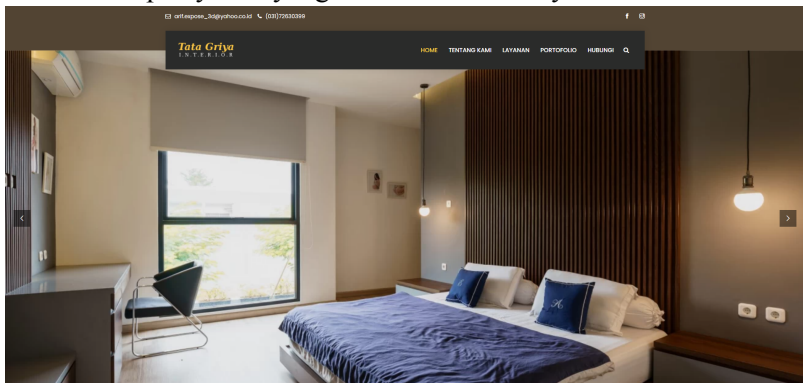


Gambar 5.2 Dashboard Admin Panel

5.3.3. View Website

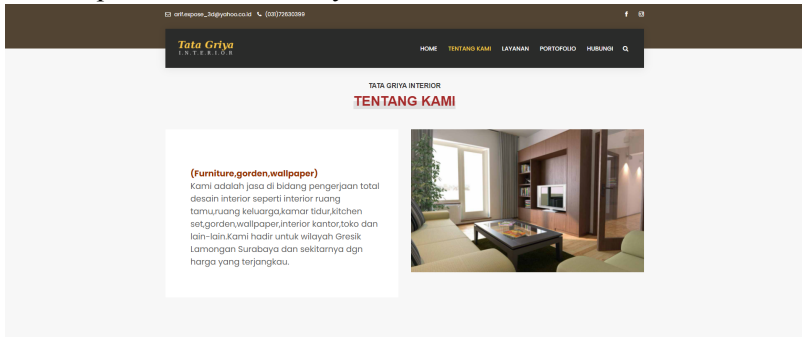
a. View Home

Gambar 5.3 berisi antarmuka halaman awal website Tata Griya Interior yang menginformasikan profil dan pelayanan yang diberikan Tata Griya Interior.



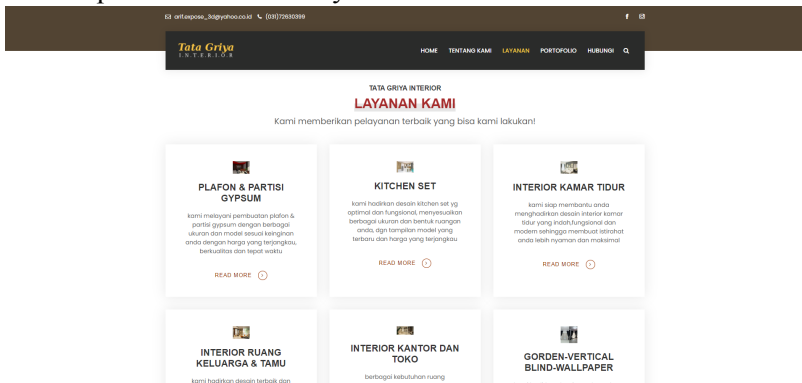
Gambar 5.3 Halaman Home

b. View Tentang Kami
 Gambar 5.4 berisi tentang informasi profil perusahaan Tata Griya Interior.



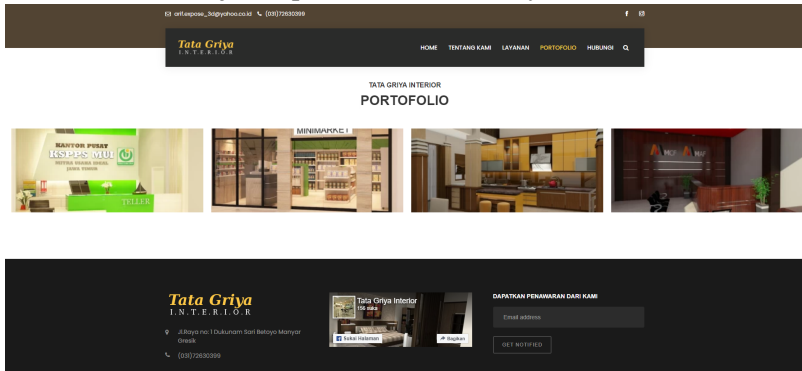
KI IFEN KAMI
 Gambar 5.4 Halaman Tentang Kami

c. View Layanan
 Gambar 5.5 berisi tentang informasi layanan perusahaan Tata Griya Interior



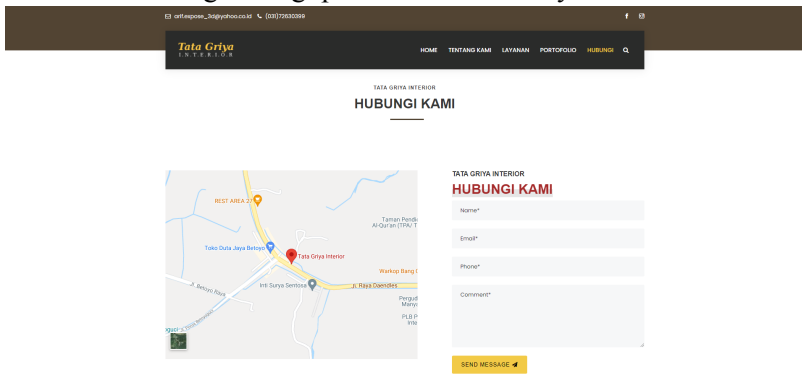
Gambar 5.5 Halaman Layanan

d. View Portofolio
Gambar 5.6 berisi tentang proyek-proyek yang telah dikerjakan perusahaan Tata Griya Interior.



Gambar 5.6 Halaman Portofolio

e. View Hubungi
Gambar 5.7 berisi tentang informasi lokasi dan cara menghubungi perusahaan Tata Griya Interior.



Gambar 5.7 Halaman Hubungi

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap Website Tata Griya Interior dan Engine Aksamedia. Pengujian dilakukan untuk memastikan kualitas website yang dibangun dan kesesuaian hasil eksekusi website dengan analisis dan perancangan perangkat lunak.

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap Website Tata Griya Interior dan Engine Aksamedia guna menguji kesesuaian dan ketepatan fungsionalitas dari seluruh sistem website.

6.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memerhatikan beberapa hasil yang diharapkan berikut ini:

- a. Kemampuan website untuk membedakan role sesuai user id pada Admin Panel.
- b. Kemampuan website untuk memberikan akses fitur sesuai role pada Admin Panel.
- c. Kemampuan website untuk memberikan setiap informasi penting kepada pengguna.
- d. Kemampuan website untuk memenuhi kebutuhan fungsionalitas setiap role, seperti mengubah Landing Page, membuat postingan blog, dan lainnya.
- e. Kemampuan aplikasi untuk memenuhi kebutuhan non-fungsional, yaitu dapat diakses di browser.

6.3. Skenario Pengujian

Skenario pengujian dilakukan dengan melakukan peran sebagai admin, superadmin dan user yang akan menjalankan fitur-fitur dan seluruh kebutuhan fungsional

dari sistem. Langkah-langkah untuk setiap kebutuhan fungsional yaitu sebagai berikut:

1. Sebagai Admin :
 - a. Admin dapat mengakses Admin Panel

2. Sebagai User :
 - a. User dapat melihat daftar layanan
 - b. User dapat melihat informasi tentang perusahaan Tata Griya Interior
 - c. User dapat melihat daftar proyek yang telah dikerjakan perusahaan Tata Griya Interior
 - d. User dapat melihat lokasi perusahaan Tata Griya Interior
 - e. User dapat menghubungi perusahaan Tata Griya Interior dengan mengirim pesan

3. Sebagai Super admin :
 - a. Super Admin dapat mengakses Admin Panel
 - b. Super Admin dapat mengatur role dan permissions pengguna
 - c. Super Admin dapat mengelola produk pada Admin Panel
 - d. Super Admin dapat mengelola portofolio pada Admin Panel
 - e. Super Admin dapat mengatur setting pada Admin Panel
 - f. Super Admin dapat mengelola landing page pada Admin Panel
 - g. Super Admin dapat mengelola blog pada Admin Panel
 - h. Super Admin dapat mengelola galleries pada Admin Panel

6.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku website Tata Griya Interior terhadap kasus skenario uji coba. Pengujian dilakukan oleh pihak pengembang, pengguna, dan pembimbing lapangan.

Pengujian dibagi menjadi 3 role user. Tabel di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

1. Sebagai Admin

Tabel 6. 1 Evaluasi Pengujian sebagai Admin

| No. | Kriteria Pengujian | Hasil Pengujian |
|------------|-----------------------------------|------------------------|
| 1 | Admin dapat mengakses Admin Panel | Terpenuhi |

2. Sebagai User

Tabel 6. 2 Evaluasi Pengujian sebagai User

| No. | Kriteria Pengujian | Hasil Pengujian |
|------------|---|------------------------|
| 1 | User dapat melihat daftar layanan | Terpenuhi |
| 2 | User dapat melihat informasi tentang perusahaan Tata Griya Interior | Terpenuhi |
| 3 | User dapat melihat daftar proyek yang telah dikerjakan perusahaan Tata Griya Interior | Terpenuhi |
| 4 | User dapat melihat lokasi perusahaan Tata Griya Interior | Terpenuhi |
| 5 | User dapat menghubungi perusahaan Tata Griya Interior dengan mengirim pesan | Terpenuhi |

3. Sebagai Super Admin

Tabel 6. 3 Evaluasi Pengujian sebagai Super Admin

| No. | Kriteria Pengujian | Hasil Pengujian |
|-----|---|-----------------|
| 1 | Super Admin dapat mengakses Admin Panel | Terpenuhi |
| 2 | Super Admin dapat mengatur role dan permissions pengguna | Terpenuhi |
| 3 | Super Admin dapat mengelola produk pada Admin Panel | Terpenuhi |
| 4 | Super Admin dapat mengelola portofolio pada Admin Panel | Terpenuhi |
| 5 | Super Admin dapat mengatur setting pada Admin Panel | Terpenuhi |
| 6 | Super Admin dapat mengelola landing page pada Admin Panel | Terpenuhi |
| 7 | Super Admin dapat mengelola blog pada Admin Panel | Terpenuhi |
| 8 | Super Admin dapat mengelola galleries pada Admin Panel | Terpenuhi |

Dengan hasil pengujian pada tabel di atas, dapat disimpulkan bahwa secara keseluruhan fitur telah memenuhi kriteria-kriteria yang sudah disebutkan pada bagian-bagian sebelumnya.

6.5. Evaluasi Oleh Dosen Pembimbing

6.5.1. SDLC

Pada pengerjaan website ini, seharusnya menerapkan siklus pengembangan sistem atau yang biasa disebut *Systems Development Life Cycle* (SDLC). SDLC digunakan dalam langkah awal membangun sistem agar sistem yang dibangun berjalan sesuai harapan. SDLC sendiri terdiri dari beberapa tahapan, antara lain :

a. *Planning*

Pada tahap ini, yang dilakukan adalah merencanakan berbagai *requirements* yang diperlukan dalam membangun website seperti *framework* yang digunakan, *tools* yang digunakan, serta biaya.

b. *Analysis*

Pada tahap ini, yang dilakukan adalah analisis sistem. Analisis yang dilakukan meliputi yaitu identifikasi masalah dan peluang yang dituju oleh website yang akan kami buat. Hasil yang didapat adalah kami harus membuat ulang website company profile Tata Griya Interior dikarenakan website yang lama sudah termakan oleh zaman, khususnya pada poin UI dan UX nya. Tujuan dari kami membuat ulang website Tata Griya Interior selain untuk memperbarui tampilan adalah juga untuk sarana promosi perusahaan Aksamedia sebagai jasa pembuatan website.

c. *Design*

Pada tahap ini, yang dilakukan adalah merancang design website yang akan dibangun. Kami berdiskusi dan merancang design dengan memperhatikan kekurangan dari UI/UX website yang lama. Hasilnya adalah kami menggunakan framework CSS yaitu Bootstrap dan mengedit beberapa CSS nya sesuai kreativitas masing-masing dan kesepakatan bersama.

Pada tahap pengerjaan website ini, juga seharusnya dibuat suatu surat persetujuan atau perjanjian yang berisi detail rancangan website yang nantinya dimuat dalam website Tata Griya Interior. Fungsi dari perjanjian tersebut menegaskan desain paten yang nantinya dibuat dalam website Tata Griya Interior, jika terdapat perubahan rancangan website maka harus dilakukan perjanjian ulang kembali. Perjanjian rancangan website dibuat dalam bentuk tertulis. Contoh surat persetujuan rancangan website adalah sebagai berikut :

(Nama Pengirim)

[Alamat]

[Kota, Kodepos]

[No Hp]

[Email]

[Tanggal surat dibuat]

Kepada Yth,

(Nama Tujuan)

[Jabatan, Nama Perusahaan]

[Alamat]

[Kota, Kodepos]

Dengan hormat,

Kami menulis surat ini bertujuan untuk pengajuan dan persetujuan rancangan desain website yang kami buat. Untuk detail rancangan bisa dilihat pada link figma berikut :

“contoh.com”

Desain ini menjadi kesepakatan bersama sehingga desain yang sudah disepakati dan ditandatangani tidak bisa dilakukan revisi untuk kedepannya.

Demikian surat pengajuan yang kami sampaikan, atas kesempatan dan perhatiannya kami ucapkan terima kasih.

Pemohon,

Menyetujui, Pembimbing Lapangan

[TTD]

[TTD]

(Nama Pengirim)

(Nama Pembimbing Lapangan)

d. Implementation

Disini adalah tahap dimana kami memulai untuk membuat website company profile tersebut. Pertama kita membuat admin panel terlebih dahulu, namun sebagian dari admin panel sudah dikerjakan terlebih dahulu oleh pihak Aksamedia, dan kami ditugaskan untuk melanjutkannya. Admin panel ini nantinya sebagai *engine* untuk membuat website company profile. Di admin panel kita bisa menginput blog, gallery, slider, about, dsb yang berkaitan dengan fitur company profile

e. Testing & Integration

Pada tahap ini kami melakukan pengujian terhadap admin panel dan website yang telah dibuat. Disini kami mendapatkan beberapa error, namun kami berhasil mengatasi error tersebut dengan bimbingan pihak Aksamedia. Setelah mengatasi semua error dan program berjalan lancar, kami mengirim program yang telah kami buat ke server Aksamedia melalui FTP (File Transfer Protocol) untuk dihosting pihak Aksamedia

f. Maintenance

Pada tahap ini, yang dilakukan adalah pemeliharaan sistem dan update sistem. Hal ini dilakukan bertujuan untuk meminimalisir error karena perkembangan waktu dan dapat berjalan dengan optimal. Tahap ini dilakukan oleh pihak Aksamedia.

6.5.2. Metode Perancangan UX

Pada pengerjaan website ini juga, seharusnya menerapkan Metode UX (User Experience). Metode UX untuk pemecahan suatu permasalahan yang berhubungan dengan interaksi antara pengguna dengan sistem aplikasi. Metode UX sendiri terdiri dari beberapa tahapan, antara lain :

a. User centered design

Metode user centered design atau desain yang berpusat pada pengguna, berfokus pada pendekatan melalui proses diskusi dengan pengguna langsung. Biasanya akan dilakukan proses wawancara oleh UX Researcher. Dimana dari proses dialog tersebut, diharapkan mampu memberikan informasi berupa data yang dapat dijadikan pedoman untuk pembuatan sebuah desain. Sehingga, desain produk awal yang akan dibuat lebih berfokus pada masukan atau saran yang diberikan oleh beberapa user yang telah diwawancarai.

b. Activity centered design

Metode activity centered design atau desain yang berpusat pada aktivitas, berfokus pada pembuatan desain aplikasi tergantung dari aktivitas pada organisasi atau perusahaan yang ditempati. Dimana setiap perusahaan, memiliki pondasi dan sistem kerja yang berbeda – beda. Aktivitas disini terbagi menjadi task (tugas), actions (tindakan), dan operations (operasi). Ketiga jenis aktivitas tersebut yang akan membantu dalam membuat struktur UX design yang baik dan sesuai dengan produk yang akan dibuat.

c. *Simple design*

Metode simple design atau desain sederhana, adalah pembuatan sebuah desain yang mementingkan kesederhanaan dan tidak mengandung unsur kerumitan di setiap elemen. Jadi, metode ini lebih mengutamakan dalam pembuatan desain yang sesederhana mungkin, namun memiliki tampilan yang baik, serta mendukung pengalaman pengguna yang efektif.

d. *Goal Directed Design*

Metode goal directed design atau desain sebagai fokus utama, dari pengertian tersebut dapat disimpulkan bahwa, pada metode ini dapat dilakukan wawancara ke beberapa pengguna yang nantinya data tersebut diolah langsung ke dalam sebuah desain aplikasi.

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN

Kesimpulan yang didapat setelah melakukan pembuatan website Tata Griya Interior pada kegiatan kerja praktik di perusahaan Aksamedia adalah sebagai berikut:

1. Aplikasi yang dibangun telah sesuai dengan permintaan
2. Dengan adanya aplikasi ini, dapat meningkatkan daya jual website untuk perusahaan Aksamedia.
3. *Framework* yang digunakan pada proses pembuatan aplikasi adalah *Laravel* dimana *framework* yang cukup umum digunakan dalam pengembangan web, sehingga apabila ingin dilakukan *maintenance* dapat dilakukan lebih mudah.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- | | |
|-----|---|
| [1] | "Visual Studio Code"[Online]. Available : https://code.visualstudio.com/docs [Accessed 5 Februari 2021] |
| [2] | XAMPP for OS X, Linux and Windows [Online]. Available : https://www.apachefriends.org/index.html [Accessed 5 Februari 2021] |
| [3] | Lardinois, F. (2015). Microsoft Launches Visual Studio Code, a Free CrossPlatform Code Editor for OS X, Linux and Windows. [Accessed 5 Februari 2021] |

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS I

Nama : Gema Adi Perwira
Tempat, Tanggal Lahir : Bangkalan, 5 Oktober 1999
Jenis Kelamin : Laki-laki
Agama : Islam
Status : Belum Menikah
Alamat Asal : Zona Neighbourhood NA25, Kec. Pakis,
Kab. Malang
Alamat Surabaya : Jl. Keputih Tegal Timur 3C No 31
Telepon : 081233256191
Email : gemaadi9@gmail.com

PENDIDIKAN FORMAL

2018 – sekarang : Mahasiswa S1 Informatika ITS
2015 – 2018 : SMA Negeri 5 Malang
2012 – 2015 : SMP Negeri 21 Malang
2006 – 2012 : SD Negeri Lesanpuro 4 Malang

KEMAMPUAN

- *Web Programming* (HTML, PHP, CSS, JavaScript)
- *Programming* (C, C++, Python)
- *Database Manajemen* (Oracle, MySQL)
- *Software* Perkantoran (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Departemen Informatika, Fakultas Teknologi
Elektro dan Informatika Cerdas, Institut
Teknologi Sepuluh Nopember Surabaya
Angkatan : 2018
Semester : 6 (Lima)
IPK : 3.25 (Semester 5)

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS II

Nama : Nodas Uziel Putra Serpara
Tempat, Tanggal Lahir : Jayapura, 12 Februari 2000
Jenis Kelamin : Laki-laki
Agama : Protestan
Status : Belum menikah
Alamat Asal : Jl. Kemiri Padang Bulan
Alamat Surabaya : Jl. Keputih Tegal Timur 3C No 31
Telepon : 081383206587
Email : uzielnodas@gmail.com

PENDIDIKAN FORMAL

2018 – sekarang : Mahasiswa S1 Informatika ITS
2015 – 2018 : SMAN 3 Jayapura
2012 – 2015 : SMP Kristen BAIK
2006 – 2012 : SD Kristen BAIK

KEMAMPUAN

- *Web Programming* (HTML, PHP, CSS, JavaScript)
- *Programming* (C, C++, Python)
- *Database Manajemen* (Oracle, MySQL)
- *Software* Perkantoran (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Departemen Informatika, Fakultas
Teknologi Elektro dan Informatika
Cerdas, Institut Teknologi Sepuluh
Nopember Surabaya
Angkatan : 2018
Semester : 6 (Lima)
IPK : 3.30 (Semester 5)

[Halaman ini sengaja dikosongkan]