

Adaptasi Model SDLC Iteratif Terhadap Pendekatan *Mobile-first* Untuk Pengembangan Antarmuka *Web Responsive*

Aswar Hanif¹, Markus Lilik Martanto², Harna Adianto³

Jurusan Teknologi Komputer, Fakultas Teknik & Informatika, Universitas Bina Sarana Informatika^{1,2}

Jurusan Teknik Elektro, Fakultas Teknik & Informatika, Universitas Bina Sarana Informatika³

Jl. Kramat Raya No. 98, Kwitang, Kec. Senen, Jakarta Pusat

E-mail : aswar.awf@bsi.ac.id¹, markus.mkk@bsi.ac.id², harna.hho@bsi.ac.id³

Abstract -- Online products and media have followed the market by accommodating user interfaces of mobile equipment. Therefore, the development of website interfaces must meet the access requirements of different devices. The mobile-first approach starts a web page development from the smallest screen sizes to larger screens. With the majority of internet access come from mobile equipment, the mobile-first approach could be a more suitable choice. The iterative SDLC model, which allows a project to be built in chunks or per iteration, is adapted to the mobile-first approach, by making the user interface development for a range of screen sizes, as an iteration. A simple responsive website was built using this adaptation development method. The development consists of 3 iterations for three ranges of screen sizes. Mobile, tablet and desktop. This method requires a product that can meet the requirements and pass the testing stages properly at each iteration. So that when it went through the last iteration, the product would have gone through 3 times of testing and reviews. In the end, the application of this method shows satisfactory results. If success is measured by the end product, then this adaptation method experiment is arguably quite successful.

Keywords: iterative, mobile-first, interface, web, responsive

Abstraksi -- Saat ini produk dan media informasi di internet sudah mengikuti perkembangan pasar digital untuk mengakomodasi antarmuka pengguna peralatan seluler. Karena itu pengembangan antarmuka situs web harus memenuhi kebutuhan akses dari peralatan-peralatan yang berbeda. Pendekatan *mobile-first* memulai pengembangan halaman web dari ukuran tampilan layar terkecil, dan akhirnya untuk tampilan layar besar. Dengan jumlah mayoritas akses internet menggunakan peralatan seluler, pendekatan *mobile-first* bisa lebih sesuai. Model SDLC iteratif, yang memungkinkan proyek untuk dibangun per bagian atau per iterasi, diadaptasi terhadap pendekatan *mobile-first*, dengan menjadikan pembuatan antarmuka untuk sebuah jangkauan ukuran tampilan layar, sebagai satu iterasi. Sebuah situs web *responsive* sederhana dibangun menggunakan metode pengembangan adaptasi ini. Tahap pengembangan ditentukan menjadi 3 iterasi untuk tiga jangkauan ukuran layar, yaitu *mobile*, tablet, dan *desktop*. Metode ini menuntut dihasilkannya sebuah situs web yang bisa memenuhi analisis kebutuhan serta melalui pengujian dengan baik pada tiap iterasinya. Sehingga saat sudah melewati iterasi terakhir, produk sudah melalui 3 kali pengujian dan review. Pada akhirnya, penerapan metode ini menunjukkan hasil produk yang cukup memuaskan. Jika dilihat dari produk yang dihasilkan, uji coba metode adaptasi ini bisa dibilang cukup berhasil.

Kata Kunci: iteratif, mobile-first, antarmuka, web, responsive

I. PENDAHULUAN

Teknologi seluler tengah menyebar dengan cepat di seluruh dunia. Menurut survey, median data kepemilikan *smartphone* di negara dengan ekonomi maju adalah 76%, dan di negara berkembang adalah 45%. [1] Pada tahun 2020, data statistik menunjukkan jumlah pengguna *smartphone* di Indonesia diperkirakan mencapai lebih dari 81 juta pengguna. [2] Saat ini produk dan media informasi di internet, seperti situs web, sudah mengikuti perkembangan pasar digital untuk mengakomodasi antarmuka pengguna agar bisa ditampilkan dengan baik pada peralatan seluler. Tapi, meskipun akses internet sebesar 60% menggunakan *smartphone*, 22% masih menggunakan laptop atau *desktop*, dan 8% menggunakan tablet. [3] Jadi meskipun dengan popularitas *smartphone* yang tinggi, konsumen juga memiliki peralatan lain seperti *desktop* dan tablet yang mereka gunakan untuk pembelian dan peninjauan yang lebih mendetail. [4] Karena itu pengembangan antarmuka situs web harus memenuhi kebutuhan akses dari peralatan-peralatan yang berbeda ini.

Desain web responsif adalah penggunaan *cascading style sheets* (CSS) untuk secara fleksibel memodifikasi tampilan halaman web sesuai dengan *viewport* alat yang digunakan. [5] Pendekatan yang menggunakan *graceful degradation* ini adalah yang lebih umum digunakan sekarang. Desain responsif dimulai dari *desktop*, yaitu ukuran resolusi terbesar, dan kemudian bertahap turun ke tampilan layar terkecil. Jadi meskipun tata letak dan konten bisa

dimuat di tampilan layar *smartphone*, navigasi, konten, dan kecepatan *download*, tetap lebih sesuai dengan situs *web* tradisional. [6] Alternatif untuk desain *web* responsif, adalah pendekatan *mobile-first*. Istilah *mobile-first* mulai mendapatkan perhatian sejak Google mengumumkan bahwa mereka mengadopsi pola pikir ini, pada tahun 2010. Pendekatan *mobile-first* menggunakan *progressive enhancement*, yang berarti pembuatan halaman *web* dimulai untuk tampilan layar terkecil, dan akhirnya untuk tampilan layar besar. [7] Menggunakan pendekatan ini, dasar dari desain tampilan antarmuka secara otomatis akan mendahulukan *mobile user-experience*. Dan dengan jumlah mayoritas akses internet menggunakan peralatan seluler, membangun antarmuka situs *web* dengan pendekatan *mobile-first* bisa lebih sesuai.

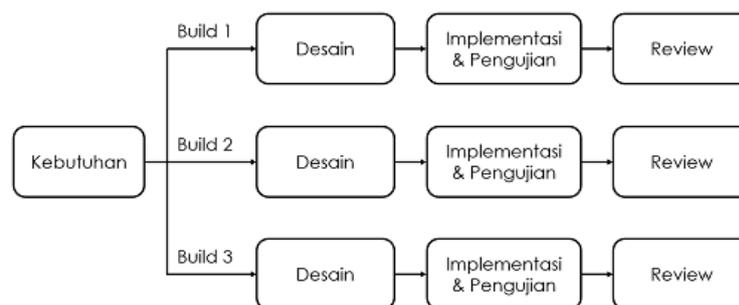
Pengembang perangkat lunak dan *web* menggunakan *software development life cycle* (SDLC) untuk membangun proyek dengan sukses. Bermula pada tahun 1960-an, secara umum tahap-tahap SDLC tidak banyak berubah. Tahap-tahap ini masih berisi perencanaan terinci yang menggambarkan pengembangan, perawatan, dan peningkatan yang berhubungan dengan sebuah perangkat lunak. [8] Ada berbagai variasi model SDLC, yang masing-masing memiliki kelebihan dan kekurangan, serta mengikuti urutan tahap yang unik untuk tipenya. [9]

Salah satu tipe model SDLC adalah model iteratif, yang memungkinkan proyek untuk dibangun per bagian (iterasi), di mana setiap iterasi berikutnya berisi fungsi-fungsi tambahan. [10] Model SDLC iteratif ini diharapkan bisa diadaptasi untuk pendekatan *mobile-first*, dengan menjadikan pembuatan antarmuka untuk tiap ukuran tampilan layar, sebagai sebuah iterasi.

Villa Sinergi adalah sebuah usaha penginapan di area Kota Batu, Malang. Penginapan yang pembukaannya tertunda karena pandemi covid-19 ini ingin mempersiapkan kegiatan pemasarannya, di mana salah satunya adalah pemasaran digital dengan memiliki sebuah situs *web*. Situs *web* masih merupakan alat pemasaran daring terpenting, karena semua media pemasaran lainnya seperti media sosial, pemasaran email, brosur, blog, dan sebagainya, merujuk ke situs *web*. [11] Sebuah situs *web* sederhana untuk Villa Sinergi akan dibangun untuk bisa digunakan sebagai alat pemasarannya. Pembangunan situs *web* ini juga akan digunakan sebagai studi kasus pengembangan antarmuka *web* menggunakan adaptasi model SDLC iteratif terhadap pendekatan *mobile-first*.

II. METODOLOGI PENELITIAN

Perekayasa perangkat lunak harus mengikuti tahap-tahap SDLC dalam membangun produk perangkat lunak yang terakayasa dengan baik untuk pengguna, dan untuk pertimbangan faktor-faktor yang mempengaruhi kualitas. [12] Model iteratif memisahkan kemajuan produk dari sebuah aplikasi menjadi bagian-bagian kecil. Fitur model iteratif adalah akan adanya sebuah model sistem yang bekerja pada bagian awal kegiatan pengembangan, dan ini membuat penemuan kesalahan menjadi lebih mudah. [13]

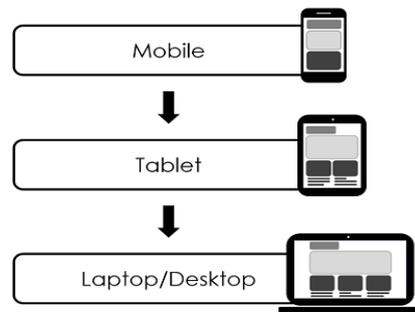


Gambar 1. Model SDLC Iteratif

Model SDLC iteratif terdiri dari pengulangan empat tahap yang disebutkan di bawah ini, sebagai sebuah urutan. [14]

1. Tahap Kebutuhan: Informasi terkait sistem dikumpulkan dan dianalisis. Kebutuhan yang dikumpulkan kemudian direncanakan sesuai untuk pengembangan sistem.
2. Tahap Desain: Solusi perangkat lunak disiapkan untuk memenuhi kebutuhan desain. Desain sistem mungkin yang baru atau perpanjangan dari yang dibangun sebelumnya.
3. Implementasi dan Pengujian: Sistem dikembangkan dengan mengkodekan dan membangun antarmuka pengguna dan modul yang kemudian dimasukkan dan diuji.
4. Tahap Review: Dimana perangkat lunak diestimasi dan diperiksa sesuai kebutuhan saat ini. Kemudian, kebutuhan lebih lanjut didiskusikan dan ditinjau untuk mengusulkan pembaruan di iterasi berikutnya.

Pendekatan *mobile-first* menggunakan *progressive enhancement*, yang berarti pembuatan halaman *web* dimulai untuk tampilan layar terkecil hingga tampilan layar terbesar. Jumlah ukuran tampilan bisa disesuaikan dengan kebutuhan, namun secara umum dibagi tiga, ukuran tampilan *smartphone*, tablet, dan *desktop* atau *laptop*.

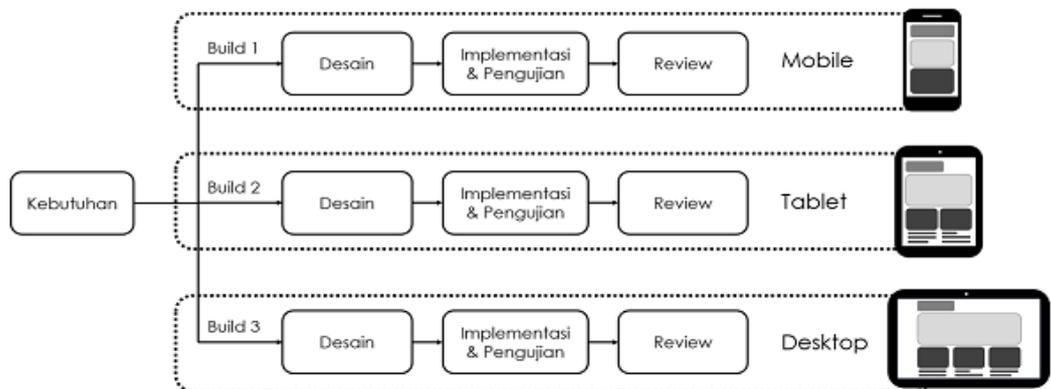


Gambar 2. Pendekatan Desain Web *Mobile-first*

Situs *web* akan dibangun dengan mencoba mengadaptasi model SDLC iteratif terhadap pendekatan *mobile-first*, dengan menjadikan pembuatan antarmuka untuk tiap ukuran tampilan layar, sebagai sebuah iterasi. Setiap iterasi umumnya adalah sebuah proses miniatur dari model *Waterfall*, dengan umpan balik dari tiap tahap yang menyediakan informasi yang dibutuhkan untuk tahap berikutnya. [15] Meskipun banyak variasi ukuran *viewport*, untuk kesederhanaan, pembangunan antarmuka akan dibagi menjadi 3 kelompok ukuran sebagai berikut:

- *Mobile* – Untuk alat dengan ukuran lebar tampilan < 768px
- *Tablet* – Untuk alat dengan ukuran lebar tampilan 768px hingga < 1200px
- *Laptop/desktop* – Untuk alat dengan ukuran lebar tampilan 1200px ke atas.

Dan sesuai model iterasi, setiap ukuran tampilan akan melalui tahap desain, implementasi, pengujian, dan review, untuk kemudian semua tahap itu diulangi lagi pada iterasi berikutnya yaitu ukuran tampilan yang lebih besar.



Gambar 3. Adaptasi Model Iteratif terhadap pendekatan *Mobile-first*.

III. HASIL DAN PEMBAHASAN

2.1. Analisis Kebutuhan

Kebutuhan Fungsional

Kebutuhan fungsional situs *web* yang dibangun dijelaskan pada tabel 1.

Tabel 1. Kebutuhan Fungsional

ID	Pernyataan Kebutuhan	Prioritas
KF-01	Menampilkan informasi villa secara berurutan pada halaman depan	Wajib
KF-15	Menyediakan tautan untuk ke langsung tampilan kontak pada konten <i>above the fold</i> halaman depan.	Tinggi
KF-20	Menyediakan ikon kontak yang akan otomatis terhubung dengan aplikasi whatsapp saat diklik.	Madya
KF-25	Menyediakan ikon kontak yang akan otomatis terhubung dengan aplikasi <i>email</i> saat diklik.	Rendah
KF-26	Menampilkan tautan ke media sosial villa	Tinggi
KF-30	Menyediakan halaman galeri foto	Wajib
KF-35	Menampilkan informasi mengenai tempat wisata sekitar lokasi	Wajib

Kebutuhan Konten

Informasi yang ditampilkan dibagi menjadi 3 halaman, di mana tampilan halaman depan dibagi menjadi 4 bagian kelompok informasi.

- a. Halaman Depan bagian *Overview (above the fold)*
Konten tampilan berupa gambar besar kualitas tinggi untuk menjadi cover, dengan teks sambutan yang menggambarkan villa secara umum. Pada bagian ini juga ada tautan menuju bagian Kontak. Sebuah logo akan ditempatkan pada navigasi utama.
- b. Halaman Depan bagian Fasilitas
Tampilan ini berisi teks yang menjelaskan berbagai fasilitas, gambar ukuran sedang sesuai jumlah fasilitas yang difiturkan, dan sebuah gambar ukuran besar kualitas tinggi untuk latar teks.
- c. Halaman Depan bagian Villa
Berisi 2 gambar ukuran sedang untuk mendampingi teks yang menjelaskan dua tipe villa, termasuk di dalamnya adalah akomodasi, dan harga sewa villa.
- d. Halaman Depan bagian Kontak
Konten bagian ini berisi teks informasi kontak, yaitu alamat, nomor telpon, dan email. Pada bagian ini juga ditempatkan ikon-ikon tautan kontak dan media sosial.
- e. Halaman Galeri
Halaman ini berisi gambar-gambar lokasi dengan ukuran bervariasi.
- f. Halaman Eksplorasi
Konten bagian ini adalah sejumlah gambar ukuran kecil, teks, dan tautan yang dikelompokkan untuk masing-masing area yang akan difiturkan.

Kebutuhan Kualitas

Kualitas yang harus dicapai situs *web* adalah:

- a. Kegunaan
Navigasi dan informasi yang ditampilkan pada situs *web* harus jelas dan mudah dimengerti untuk semua tingkat kemampuan pengguna.
- b. Efisiensi
Situs *web* harus dibangun dengan ukuran kecil dan ringan agar bisa diakses dengan lancar pada peralatan dengan koneksi kurang cepat.
- c. Portabilitas
Situs *web* harus bisa menyesuaikan tampilannya dengan peralatan-peralatan yang memiliki berbagai ukuran tampilan yang berbeda.

Kebutuhan Lingkungan Sistem

- a. Aplikasi peselancar internet yang mendukung HTML5, CSS3, dan jQuery.
- b. Peralatan terhubung internet yang memiliki layar dengan lebar *viewport* minimal 300px.
- c. *Web* hosting.

Kebutuhan Antarmuka Pengguna

- a. Antarmuka Depan bagian *Overview (above the fold)*
Gambar utama dibuat memenuhi setengah bagian atas halaman dengan tata letak sampul untuk mengakomodasi perubahan ukuran tampilan layar. Tautan menuju bagian kontak harus dibuat jelas dan kontras dengan latar. Navigasi diletakkan paling atas dengan posisi tetap. Untuk ukuran *mobile*, navigasi dibuat *dropdown*. Logo diletakkan di bagian kiri.
- b. Antarmuka Depan bagian Fasilitas
Untuk keringkas tampilan, teks yang menjelaskan fasilitas dibuat terlihat hanya jika disentuh (pada layar sentuh) atau dilewati oleh penunjuk. Posisi setiap kotak yang menjelaskan fasilitas harus menyesuaikan dengan ukuran tampilan layar.
- c. Antarmuka Depan bagian Villa
Saat layar dalam jangkauan ukuran *mobile*, gambar dan teks penjelasannya ditampilkan bertumpuk. Saat layar dalam jangkauan ukuran di atas *mobile*, gambar dan teks penjelasannya ditampilkan bersebelahan, tapi tetap bertumpuk untuk tiap tipe villa.
- d. Antarmuka Depan bagian Kontak
Sebagai dekorasi, ikon-ikon dibuat berubah warna saat disentuh atau dilewati penunjuk.
- e. Antarmuka Galeri
Untuk dekorasi, dalam jangkauan ukuran *mobile* tata letak gambar secara bergantian dibuat 1 kolom dan 2 kolom. Saat layar dalam jangkauan ukuran di atas *mobile*, tata letak gambar disesuaikan estetikanya. Gambar diberikan sedikit animasi jika disentuh (pada layar sentuh) atau dilewati oleh penunjuk.

f. Antarmuka Eksplorasi

Setiap gambar, teks dan tautan dikelompokkan dalam *card*, sesuai lokasi wisatanya. Tautan dibuat kontras, dan membuka *tab* baru. Tata letak akan dibuat sederhana, untuk tampilan *mobile* 1 kolom, ukuran tablet 2 kolom, dan ukuran *desktop* 3 kolom.

Pembatas Proyek

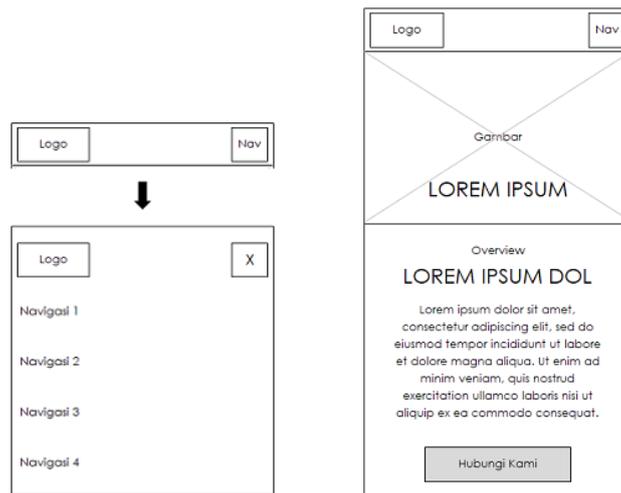
Karena pandemi covid-19, perjalanan dibatasi dan villa ditutup sehingga tidak bisa mengambil gambar-gambar baru dengan tingkat kualitas yang lebih baik.

2.2. Iterasi 1 (Mobile)

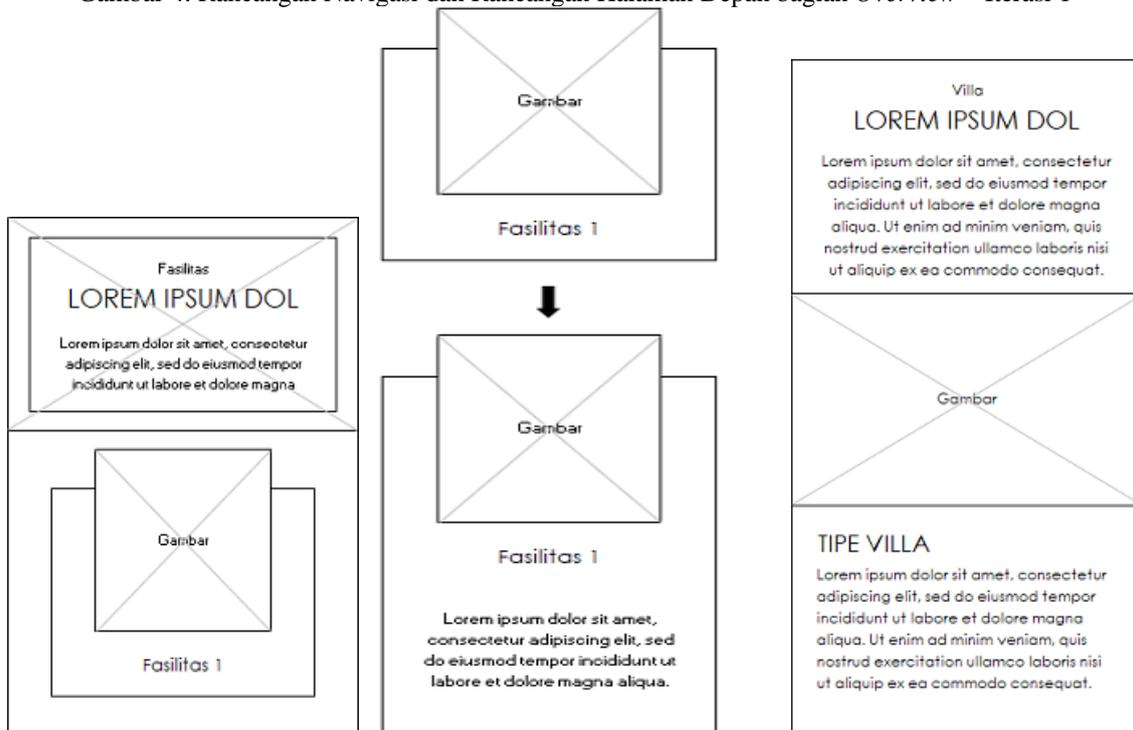
Desain Antarmuka

Situs *web* dibagi menjadi 3 halaman, di mana tampilan halaman depan dibagi menjadi 4 bagian kelompok informasi. Meskipun dalam implementasi akan disatukan secara berurutan, desain antarmuka dilakukan terpisah.

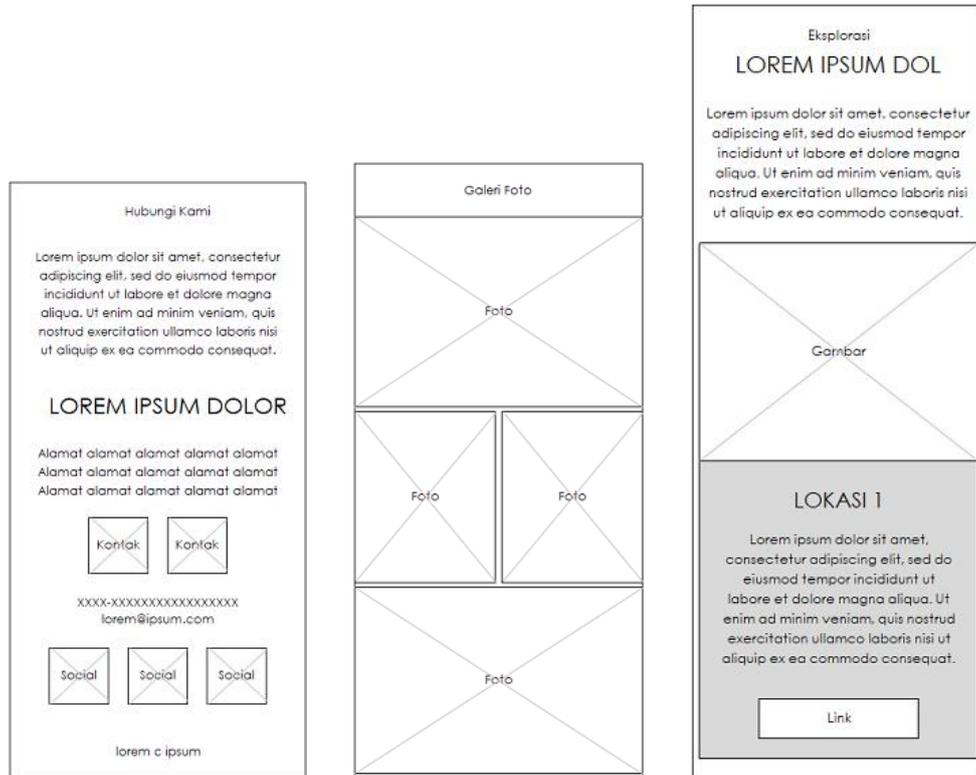
Rancangan navigasi pada situs *web* ini posisinya *fixed*. Ini maksudnya adalah bahwa navigasi akan selalu ada di posisi yang sama ke mana pun layar digerakkan, yaitu di paling atas layar tampilan.



Gambar 4. Rancangan Navigasi dan Rancangan Halaman Depan bagian *Overview* – Iterasi 1



Gambar 5. Rancangan Halaman Depan Bagian Fasilitas dan Bagian Villa – Iterasi 1



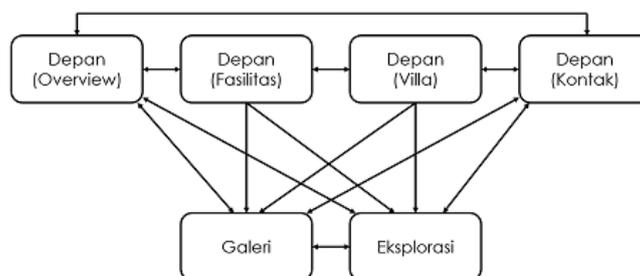
Gambar 6. Rancangan Halaman Bagian Kontak, Halaman Galeri, dan Halaman Eksplorasi – Iterasi 1

Animasi untuk bagian fasilitas dirancang agar bagian yang berisi teks judul memanjang ke bawah untuk membuat ruang bagi teks penjelasan yang muncul. Untuk bagian villa, gambar akan dibuat sedikit membesar jika disentuh (pada layar sentuh) atau dilewati oleh penunjuk sehingga memberikan kesan *zoom in*.

Ikun-ikon pada bagian kontak dirancang untuk berubah warna jika disentuh (pada layar sentuh) atau dilewati oleh penunjuk. Gambar-gambar foto pada halaman galeri akan dibuat sedikit membesar dan diberikan filter *grayscale* jika disentuh (pada layar sentuh) atau dilewati oleh penunjuk, sebagai pemberi kesan animasi.

Desain Struktur Navigasi

Rancangan struktur navigasi adalah campuran antara linier dan non-linier. Empat bagian pada halaman depan berurutan secara linier. Karena rancangan navigasinya yang bersifat *fixed*, bagian *overview*, bagian kontak, halaman galeri, dan halaman eksplorasi membuatnya menjadi non-linier. Bagian fasilitas dan bagian villa tidak bisa dicapai langsung dari halaman galeri dan halaman eksplorasi, harus melalui bagian *overview* atau bagian kontak.

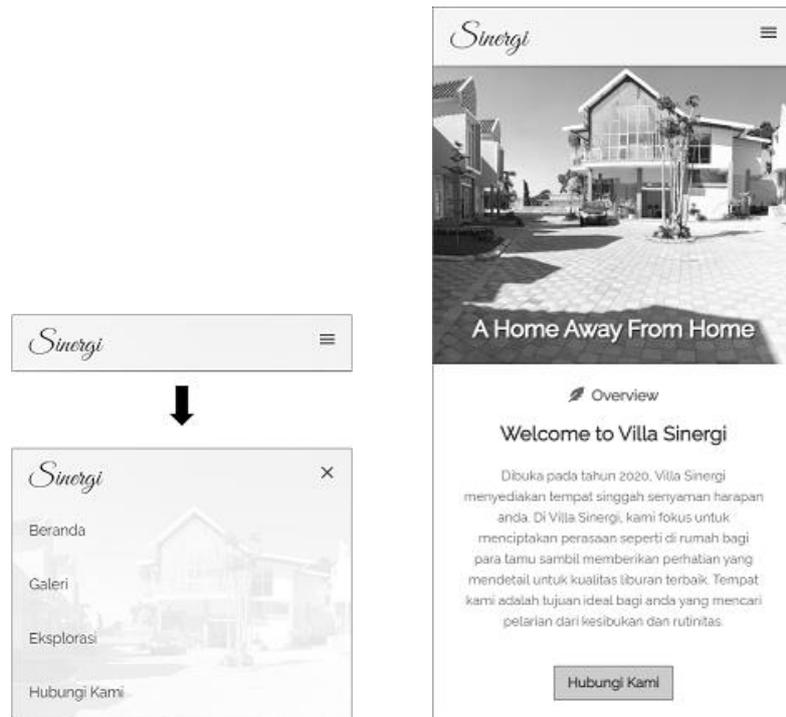


Gambar 7. Rancangan Struktur Navigasi

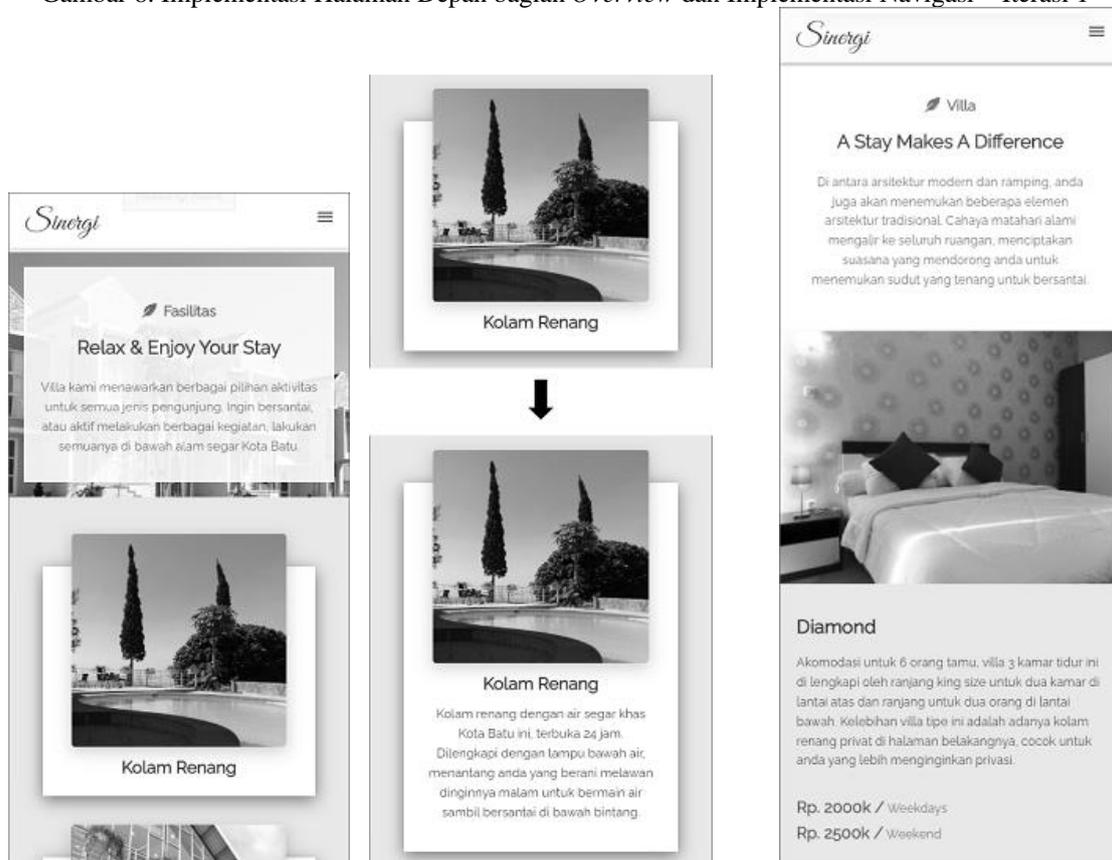
Implementasi

Implementasi dilakukan sesuai rancangan dengan pemilihan latar utama berwarna putih untuk memberikan kesan bersih dan elegan. Logo dibuat dalam bentuk teks menggunakan *font Great Vibes*, dan semua teks dalam situs menggunakan *font Raleway*. Font-font ini berlisensi bebas dari *library Google fonts*.

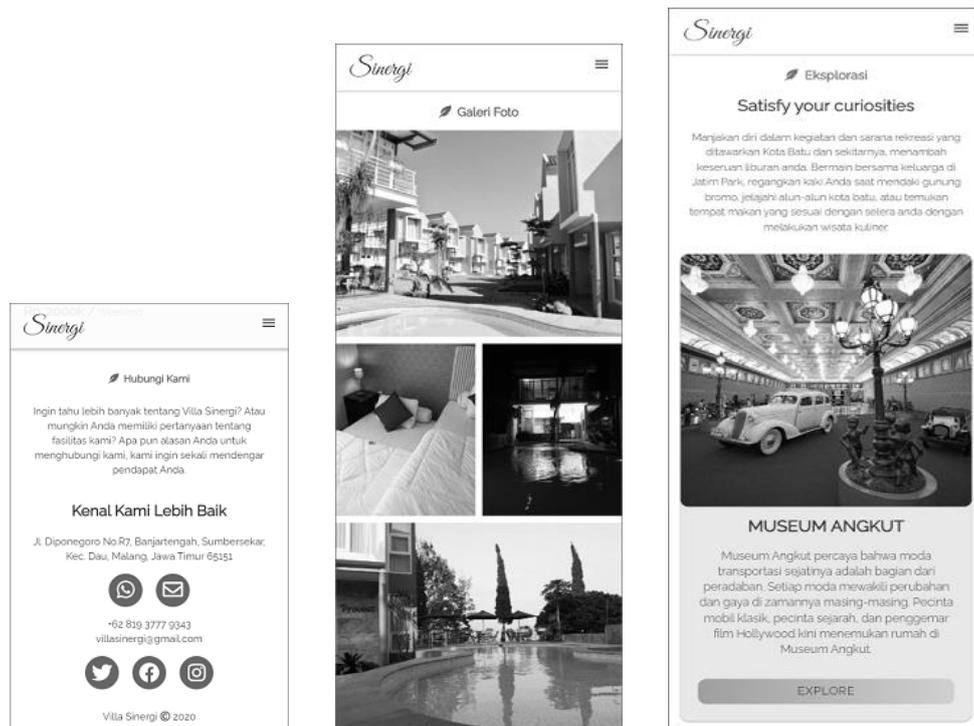
Penerapan yang tidak ada dalam rancangan adalah pemberian sebuah gambar bulu kecil di depan setiap teks bagian. Ini murni untuk estetika. Gambar ini diambil dari *library* ikon Font Awesome yang juga berlisensi bebas.



Gambar 8. Implementasi Halaman Depan bagian Overview dan Implementasi Navigasi – Iterasi 1



Gambar 9. Implementasi Halaman Depan Bagian Fasilitas dan Bagian Villa – Iterasi 1



Gambar 10. Implementasi Halaman Depan Bagian Kontak, Halman Galeri, dan Halaman Eksplorasi – Iterasi 1 Pengujian

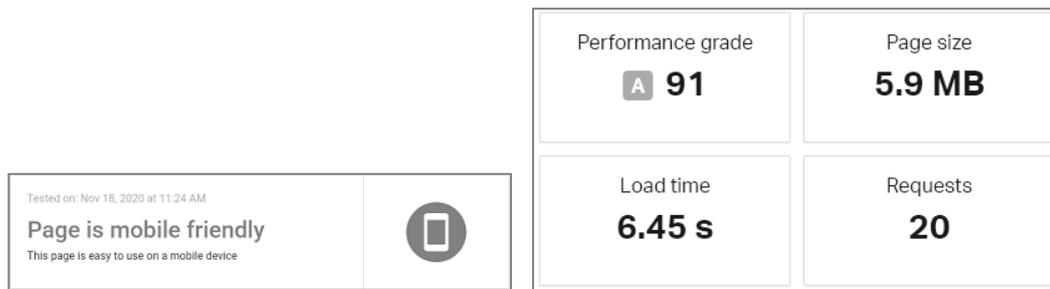
Proses pengujian pertama dilakukan dengan metode blackbox, yaitu dilihat dari masukan dan keluaran situs *web*. Proses pengujian kedua dilakukan menggunakan *Google Mobile-Friendly Test*, untuk mengetahui apakah situs *web* yang dibuat mudah digunakan pada peralatan *mobile*. Terakhir, dilakukan pengujian performa menggunakan *Pingdom Website Speed Test* untuk menguji waktu muat halaman, menganalisisnya, dan menemukan hambatan-hambatan.

Tabel 2. Hasil Pengujian Halaman Depan – Iterasi 1

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian
1	Membuka fungsi <i>dropdown</i> pada navigasi utama	Tap pada ikon navigasi utama.	Situs <i>web</i> akan menghasilkan sebuah tampilan yang selalu berada di paling depan yang menampilkan daftar tautan.	Sesuai
2	Menutup menu <i>dropdown</i> pada navigasi utama	Tap pada ikon “x”, di navigasi utama.	Daftar tautan akan dihilangkan dan navigasi utama kembali menjadi ikon.	Sesuai
3	Melihat teks penjelasan pada bagian fasilitas.	Tap pada gambar, teks judul, atau kotak penampungnya.	Kotak penampung teks judul akan memanjang ke bawah untuk memberikan ruang pada teks penjelasan yang akan muncul mengambang masuk dari atas.	Sesuai
4	Membuka chat whatsapp langsung ke nomor kontak	Tap pada ikon whatsapp di bagian kontak.	Situs <i>web</i> akan menautkan ke fungsi <i>click to chat</i> whatsapp berisi pesan yang sudah disesuaikan, menuju nomor kontak yang ditentukan.	Sesuai
5	Membuka aplikasi email untuk langsung kirim ke <i>email</i> kontak.	Tap pada ikon <i>email</i> di bagian kontak.	Situs <i>web</i> akan membuka aplikasi email yang terinstal di alat pengguna untuk langsung mengirim ke alamat email kontak.	Sesuai

Tabel 3. Hasil Pengujian Halaman Galeri – Iterasi 1

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian
1	Melihat animasi pada gambar.	Tap pada pada gambar.	Gambar akan membesar dengan skala 1,5 dan diberikan filter <i>grayscale</i> , sebelum menampilkan modal yang memperlihatkan gambar foto secara keseluruhan,	Sesuai
2	Melihat gambar secara utuh.	Tap pada pada gambar.	Situs <i>web</i> akan membuka modal untuk menampilkan gambar secara utuh.	Sesuai



Gambar 11. Hasil Pengujian Google *Mobile-Friendly Test* & Pingdom *Website Speed Test* – Iterasi 1

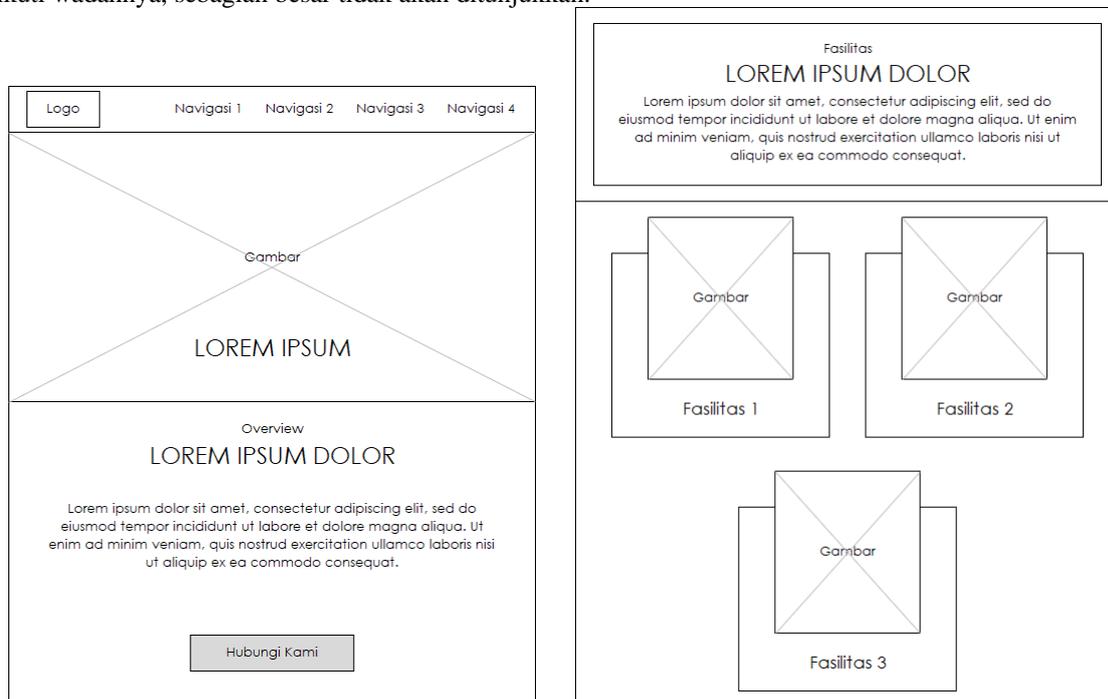
Review

Hasil implementasi pada iterasi 1 ini cukup sesuai dan memuaskan analisis kebutuhan. Halaman situs *web* ditampilkan dengan baik pada layar ukuran <768px. Hasil pengujian *blackbox* tidak menemukan masalah. Pengujian daring Google menunjukkan iterasi pertama ini *mobile friendly*, dan nilai performa 91 dari Pingdom diputuskan sudah cukup baik. Pengembangan bisa dilanjutkan ke tahap berikutnya, yaitu iterasi ke-2.

2.3. Iterasi 2 (Tablet)

Desain Antarmuka

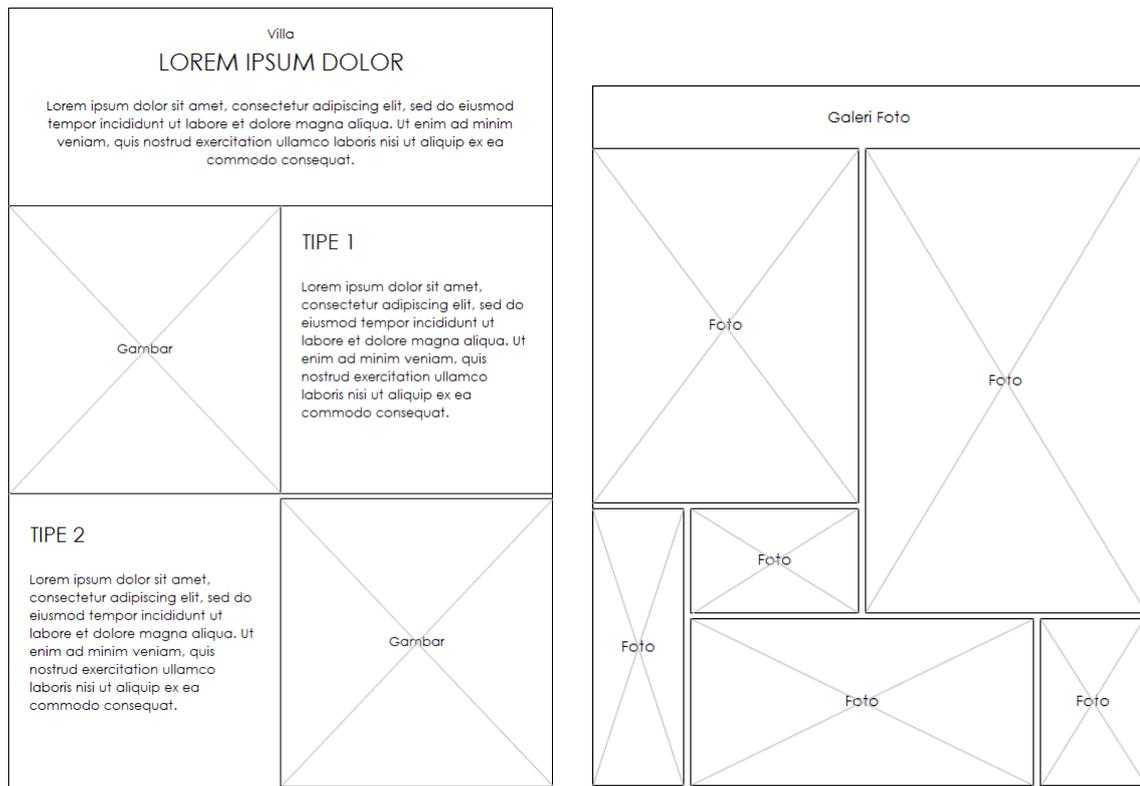
Desain antar muka yang akan ditunjukkan adalah rancangan yang dianggap menunjukkan perbedaan tata letak yang cukup signifikan dari tampilan iterasi sebelumnya. Antarmuka yang tampilannya hanya secara *fluid* mengikuti wadahnya, sebagian besar tidak akan ditunjukkan.



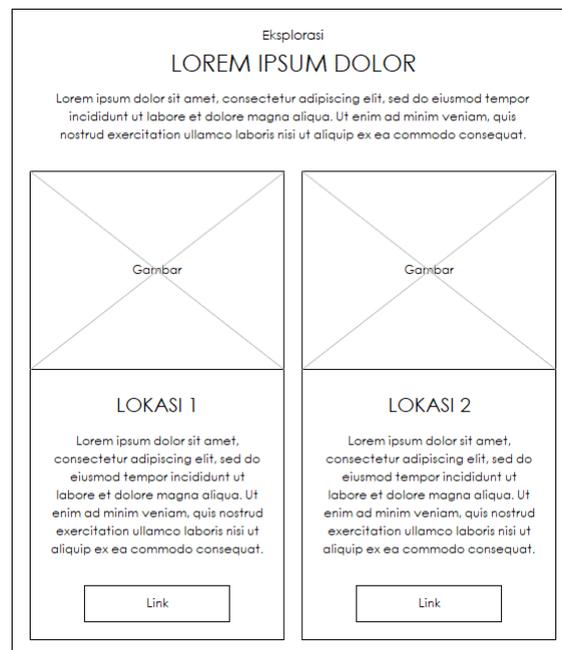
Gambar 12. Rancangan Halaman Depan bagian *Overview* dan bagian *Fasilitas* – Iterasi 2

Untuk rancangan navigasi utama, daftar tautan akan dibuat menjadi berjajar secara horizontal karena lebar tampilan layar pada iterasi ini mencukupi. Bagian fasilitas juga akan dibuat menjadi 2 kolom. Rancangan bagian villa diubah tata letaknya, di mana sebelumnya gambar dan teks penjelasan bertumpuk, perubahan lebar layar memungkinkan rancangan yang berdampingan.

Rancangan halaman galeri dibuat sedemikian rupa, murni untuk estetika. Tampilan *grid* seperti pada umumnya sebuah galeri foto, dirasa agak membosankan. Halaman eksplorasi yang sebelumnya 1 kolom, dibuat menjadi 2 kolom.



Gambar 13. Rancangan Halaman Depan bagian Villa dan Halaman Galeri – Iterasi 2



Gambar 14. Rancangan Halaman Eksplorasi – Iterasi 2

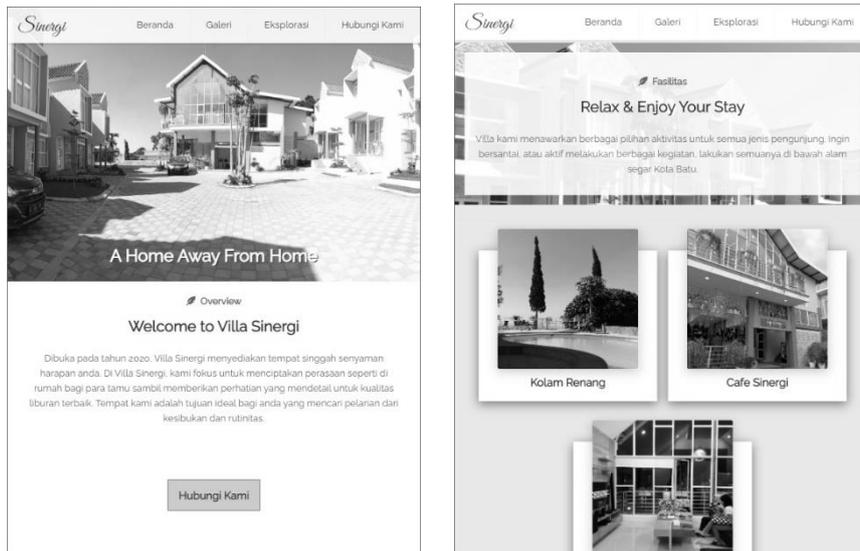
Desain Struktur Navigasi

Meskipun berubah rancangan tampilan antarmukanya, struktur navigasi tidak berubah pada iterasi ini.

Implementasi

Ada tiga teknik umum untuk mempersiapkan rancangan *web responsive*: [16]

- *Fluid grid layouts*
- Gambar dan media yang fleksibel
- *Media query* dan ukuran layar

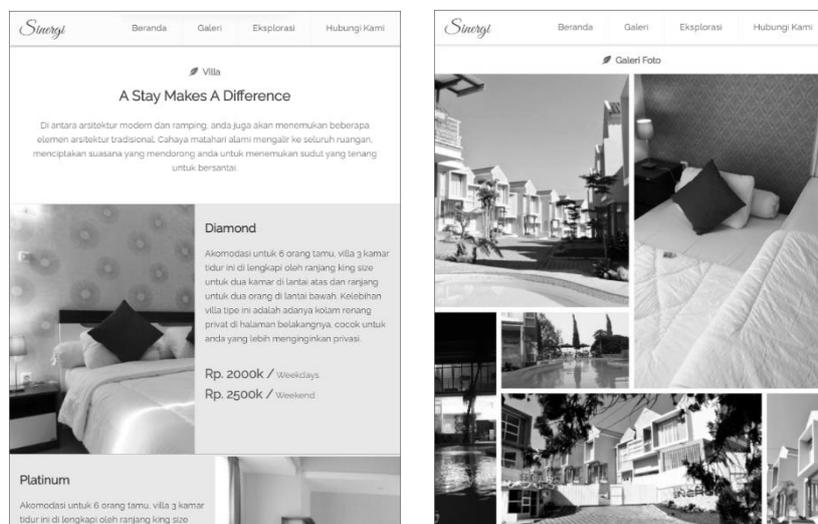


Gambar 15. Implementasi Halaman Depan bagian *Overview* dan bagian Fasilitas – Iterasi 2 Untuk navigasi utama digunakan media *query* sesuai dengan lebar layar minimal yang sudah ditentukan untuk tablet, yaitu 768px. Berikut bagian kode CSS yang memungkinkan perubahan tata letak seperti rancangan:

```
@media (min-width: 768px) {
  ....
  .header li {float: left;}
  .header .menu {clear: none; float: right; max-height: none;}
  .header .menu-icon {display: none;}
  ...
}
```

Sedangkan untuk bagian fasilitas, implementasi rancangan dicapai bukan menggunakan media *query*, tapi menggunakan fitur tampilan *flex*. Kode CSS ini sudah disiapkan untuk *responsive* sejak iterasi 1. Dia akan memenuhi baris dengan otomatis saat lebar layar mencukupi. Jadi tidak harus pada 768px, tapi kebetulan pada lebar itu dia sudah cukup untuk berdua tiap barisnya. Dan lebar maksimalnya juga sudah disiapkan pada 1100px. Sehingga dipastikan dia tidak akan memenuhi lebar layar pada iterasi 3 nanti.

```
.cards-container {
  ...
  width: 1100px; display: flex; flex-wrap: wrap;
  justify-content: center; align-items: center;
  ...
}
```



Gambar 16. Implementasi Halaman Depan bagian Villa dan Halaman Galeri – Iterasi 2

Halaman depan bagian villa, membutuhkan media *query* untuk perubahan tata letaknya mengikuti lebar layar tampilan. Fitur tampilan *flex* secara *default* akan menampilkan elemen di dalamnya bersebelahan dalam kolom-kolom. Karena gambar maupun teks penjelasannya hanya berdua di dalam *class* “*room*”, maka otomatis kedua elemen itu akan bersebelahan. Untuk Tipe villa berikutnya, dimasukkan ke dalam *class* “*room*” yang lain. Dalam potongan kode CSS juga terlihat bagaimana posisi gambar dan teks bisa bertukar.

```
@media (min-width: 768px) {
  ...
  .room {display: flex; ...}
  .room:nth-child(even) {flex-direction: row-reverse;}
  ...
}
```



Gambar 17. Implementasi Halaman Eksplorasi – Iterasi 2

Modifikasi halaman eksplorasi paling sederhana, menggunakan media *query* dan fitur tata letak terbaru CSS yaitu *grid*. Padal lebar layar 768px, tampilan dipaksa menjadi 2 bagian.

```
@media (min-width: 768px) {#eksplorasi {... display: grid; grid-template-columns: 1fr 1fr; ...}...}
```

Pengujian

Proses pengujian tidak berubah dari iterasi 1. Pertama dilakukan dengan metode *blackbox* kemudian menggunakan Google *Mobile-Friendly Test*, dan terakhir adalah pengujian performa menggunakan Pingdom *Website Speed Test*. Selain mengenai navigasi utama yang sebelumnya berbentuk *dropdown*, hasil pengujian *blackbox*, tidak berubah.

Begitu juga pengujian daring, selain perubahan insignifikan pada *load time*, hasilnya bisa dibbilang identik. Ini wajar karena perbedaan iterasi 1 dan iterasi 2 hanyalah sejumlah baris CSS yang tidak terlalu banyak.



Gambar 18. Hasil Pengujian Google *Mobile-Friendly Test* & Pingdom *Website Speed Test* – Iterasi 2

Review

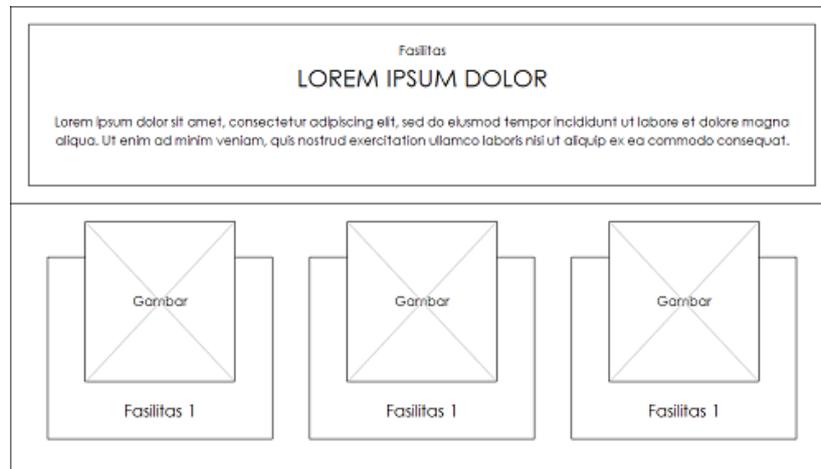
Hasil implementasi pada iterasi 2 ini, juga cukup sesuai dan memuaskan analisis kebutuhan. Halaman situs *web* ditampilkan dengan baik pada jangkauan ukuran layar tablet, yaitu 768px hingga < 1200px. Hasil pengujian

blackbox tidak menemukan masalah. Hasil pengujian daring dari Google dan Pingdom juga diputuskan sudah cukup baik. Pengembangan bisa dilanjutkan ke tahap berikutnya, yaitu iterasi ke-3.

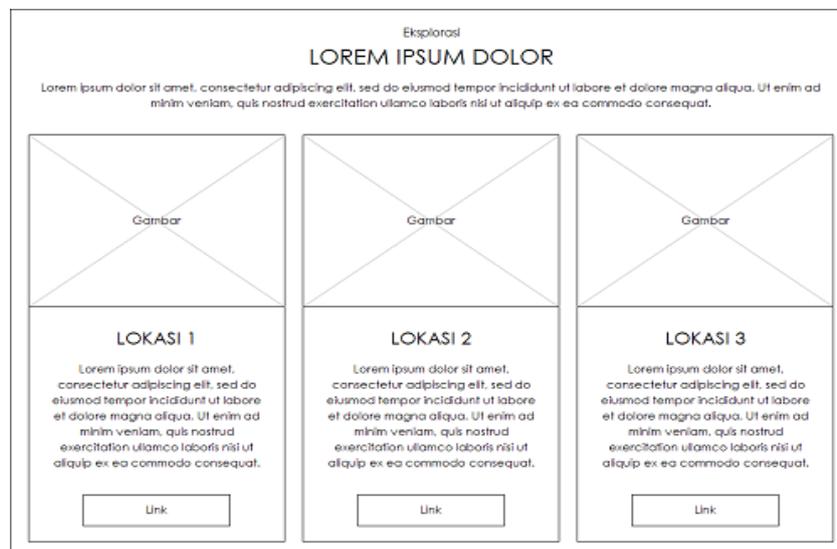
2.4. Iterasi 3 (*Desktop/laptop*)

Desain Antarmuka

Seperti iterasi pada iterasi 2, desain antar muka yang akan ditunjukkan adalah rancangan yang dianggap menunjukkan perbedaan tata letak yang cukup signifikan dari tampilan iterasi sebelumnya. Hanya 2 tampilan yang dianggap memenuhi kriteria ini.



Gambar 19. Rancangan Halaman Depan bagian Fasilitas – Iterasi 3



Gambar 20. Rancangan Halaman Eksplorasi – Iterasi 3

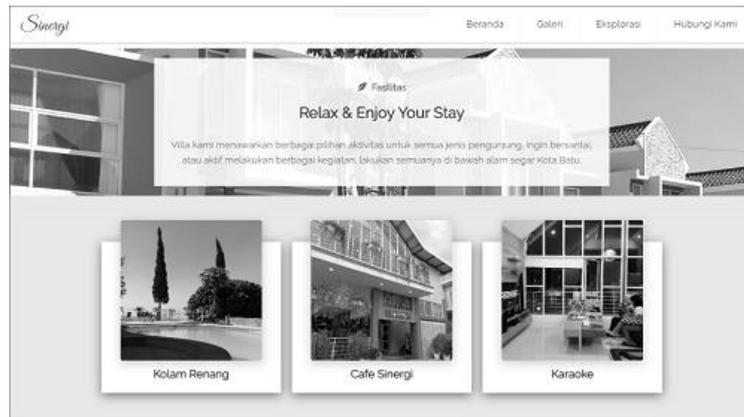
Kedua rancangan ini mirip, karena lebar layar pada iterasi ini memungkinkan, mereka akan diubah tata letaknya dari 2 kolom menjadi 3 kolom.

Desain Struktur Navigasi

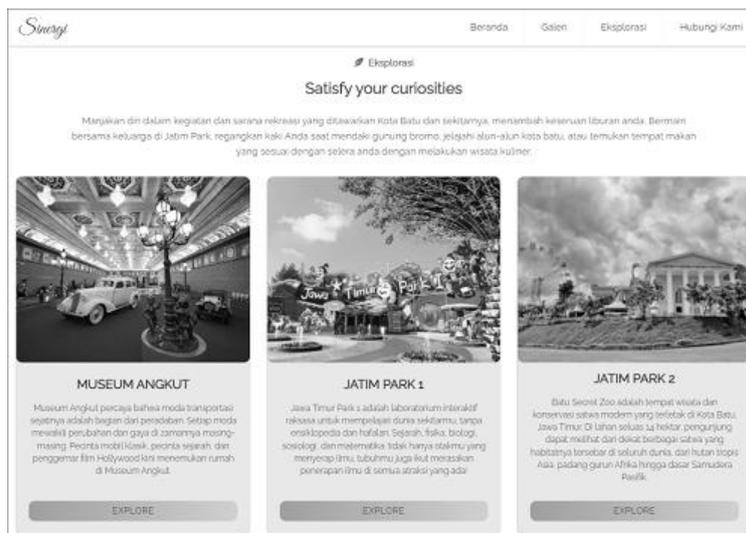
Seperti pada iterasi sebelumnya, perubahan tata letak tampilan tidak merubah struktur navigasi.

Implementasi

Untuk bagian fasilitas, sudah dijelaskan pada iterasi 2. Implementasi rancangan dicapai bukan menggunakan media *query*, tapi menggunakan fitur tampilan *flex*. Dia akan memenuhi baris dengan otomatis saat lebar layar mencukupi. Dengan ukuran layar 1200px ke atas, maka secara otomatis tiga elemen itu akan memenuhi 1 baris.



Gambar 21. Implementasi Halaman Depan bagian Fasilitas – Iterasi 3



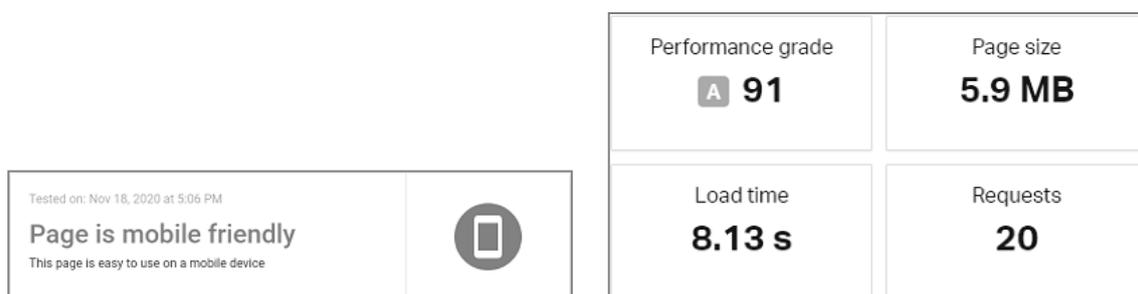
Gambar 22. Rancangan Halaman Eksplorasi – Iterasi 3

Tata letak halaman eksplorasi diimplementasikan kembali menggunakan media *query* dan fitur tata letak *grid*. Pada lebar layar minimal 1200px, tampilan dipaksa menjadi 3 bagian.

```
@media (min-width: 1200px) {
    #eksplorasi {... display: grid; grid-template-columns: 1fr 1fr 1fr; ...}}
```

Pengujian

Proses pengujian tetap sama dengan iterasi-iterasi sebelumnya. Pertama dilakukan dengan metode *blackbox* kemudian menggunakan *Google Mobile-Friendly Test*, dan terakhir adalah pengujian performa menggunakan *Pingdom Website Speed Test*. Hasil pengujian *blackbox* hanya berbeda di mana sebelumnya harus *ditap* untuk menjalankan animasi, menggunakan alat penunjuk cukup letakkan penunjuk di atas objeknya. Tidak ada perubahan lain. Pada pengujian daring, terjadi penambahan pada *load time*, tapi karena nilai performanya tetap sama, yaitu 91, maka ini sudah dianggap cukup.



Gambar 23. Hasil Pengujian *Google Mobile-Friendly Test* & *Pingdom Website Speed Test* – Iterasi 3

Review

Hasil iterasi 3 ini dianggap sudah sesuai dengan analisis kebutuhan. Dengan selesainya iterasi ke-3, halaman situs *web* bisa ditampilkan dengan baik baik pada layar *mobile*, layar tablet, maupun layar *desktop* atau *laptop*. Situs *web* sudah sepenuhnya *responsive*. Hasil pengujian daring dari Google dan Pingdom juga telah menunjukkan bahwa komabilitas dengan alat *mobile* dan performa situs *web* sudah memuaskan. Diputuskan bahwa tampilan antarmuka situs ini sudah siap untuk diluncurkan.

IV. KESIMPULAN

Penggunaan model SDLC iteratif yang diadaptasikan terhadap pendekatan *mobile-first* untuk pembangunan antarmuka situs *web* ini menghasilkan sebuah produk yang cukup baik. Pendekatan *mobile-first* memaksa pengembang untuk mendahulukan *user experience* pada antarmuka *mobile*. Dan pada kenyataannya, paling banyak waktu dialokasikan pada pembangunan antarmuka *mobile*, karena itu adalah tahap yang pertama dikerjakan. Pengaruh penerapan model iteratif terasa menekan untuk menghasilkan produk *mobile* yang sebaik-baiknya, karena proses pengembangan hanya bisa ke iterasi berikutnya setelah melalui pengujian dan review. Jadi tanpa menghasilkan sebuah situs *web mobile* yang bisa memenuhi analisis kebutuhan serta melalui pengujian komabilitas dan performa dengan baik, proses pengembangan tidak bisa berlanjut. Meskipun terasa sulit di awal, ini mempermudah iterasi-iterasi berikutnya. Proses pengujian yang berulang-ulang memperbesar kemungkinan kesalahan terdeteksi dan diperbaiki. Sehingga saat sudah melewati iterasi terakhir, produk sudah melalui 3 kali pengujian dan review.

Meskipun begitu, kekurangan-kekurangan yang pada dasarnya dimiliki model iteratif dan pendekatan *mobile-first* tetap ada. Bagian pengujian terasa redundan dan membuat proses pengembangan membosankan. Kebiasaan untuk segera membuat sebuah tampilan responsif harus ditahan, karena mengikuti ukuran yang sudah ditentukan pada iterasi yang sedang berlangsung. Fokus pada antarmuka *mobile* di awal, juga bisa membuat rancangan untuk layar yang lebih besar terasa kosong.

Pada akhirnya, penerapan metode ini menunjukkan produk yang dihasilkan cukup memuaskan. Permasalahan yang dihadapi dalam masa pengembangan mungkin bisa disimpulkan karena tidak terbiasa. Adaptasi model iteratif terhadap pendekatan *mobile-first* ini, pada dasarnya adalah sebuah metode uji coba. Jika kesuksesan dinilai dari apakah produk yang dihasilkan sesuai dengan kebutuhan, maka bisa dibilang uji coba metode ini cukup berhasil.

Penelitian lebih lanjut yang lebih baik untuk uji coba metode ini, pertama adalah menerapkannya pada proyek yang lebih besar dan lebih dinamis. Berikutnya adalah menggunakan alat pengujian yang lebih komprehensif dan mendetail untuk menghasilkan kualitas produk yang lebih baik lagi.

V. DAFTAR PUSTAKA

- [1] K. Taylor and L. Silver, "Smartphone Ownership Is Growing Rapidly Around the World, but Not Always Equally," 5 February 2019. [Online]. Available: <https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/>.
- [2] Statista Research Department, "Smartphone users in Indonesia 2011-2022," 19 February 2019. [Online]. Available: <https://www.statista.com/statistics/266729/smartphone-users-in-indonesia/>.
- [3] BOC Orenzi, "Statistik Pengguna Digital dan Internet Indonesia 2019," 24 February 2019. [Online]. Available: <https://www.boc.web.id/statistik-pengguna-digital-dan-internet-indonesia-2019/>. [Accessed September 2020].
- [4] D. Chaffey, "Mobile marketing statistics compilation," 1 September 2020. [Online]. Available: <https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>. [Accessed September 2020].
- [5] T. A. Walsh, P. McMinn and G. M. Kapfhammer, "Automatic Detection of Potential Layout Faults Following Changes to Responsive Web Pages," in *IEEE/ACM International Conference on Automated Software Engineering*, Lincoln Nebraska, 2015.
- [6] D. Kaempf, "Choose your strategy: Mobile-First Web Design vs. Responsive Web Design," 2018 March 2018. [Online]. Available: <https://darwindigital.com/mobile-first-versus-responsive-web-design/>. [Accessed September 2020].
- [7] L. Giurgiu and I. Gligorea, "Responsive Web Design Techniques," in *International conference KNOWLEDGE-BASED ORGANIZATION*, 2017.
- [8] Husson University Online, "What Is the Software Development Life Cycle?," 10 March 2016. [Online]. Available: <https://online.husson.edu/software-development-cycle/>. [Accessed September 2020].

- [9] S. Z. Iqbal and M. Idrees, "Z-SDLC Model:A New Model For Software Development Life Cycle (SDLC)," *International Journal of Engineering and Advanced Research Technology (IJEART)*, vol. 3, no. 2, pp. 1-8, 2017.
- [10] S. Barjitya, A. Sharma and U. Rani, "A detailed study of Software Development Life Cycle (SDLC) Models," *International Journal Of Engineering And Computer Science*, vol. 6, no. 7, pp. 22097-22100, 2017.
- [11] Eilis Cope, "Why Your Website Is Your Most Important Marketing Tool," 10 April 2017. [Online]. Available: <https://www.cope-salesandmarketing.com/website-important-marketing-tool/>. [Accessed September 2020].
- [12] M. K. Sharma, "A study of SDLC to develop well engineered software," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 520-523, 2017.
- [13] O. J. Okesola, A. A. Adebisi, A. A. Owoade, O. Adeaga, O. Adeyemi and I. Odun-Ayo, "Software Requirement in Iterative SDLC Model," in *Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020*, 2020.
- [14] w3schools, "SDLC Iterative Model," w3schools, [Online]. Available: <https://www.w3schools.in/sdlc-tutorial/iterative-model/>. [Accessed September 2020].
- [15] S. Nugroho, S. H. Waluyo and L. Hakim, "Comparative Analysis of Software Development Methods between Parallel, V-Shaped and Iterative," *International Journal of Computer Applications*, vol. 169, no. 11, pp. 7-11, 2017.
- [16] W. I. Bader and A. I. Hammouri, "Responsive Web Design Techniques," *International Journal of Computer Applications*, vol. 150, no. 2, pp. 18-27, 2016.