

# PENGIMPLEMENTASIAN APLIKASI GENERATOR KODE HTML DAN CSS UNTUK PERANCANGAN WEB RESPONSIF

Altiano R. Gerung<sup>1</sup>, Liza Wikarsa<sup>1</sup>, Rinaldi Munir<sup>2</sup>

<sup>1</sup>Program Studi Teknik Informatika; Fakultas Teknik

<sup>2</sup>Program Studi Teknik Informatika; Sekolah Teknik Elektro dan Informatika (STEI) ITB

<sup>1</sup>Universitas Katolik De La Salle Manado; Kombos – Kairagi 1 Manado, Telp (0431) 871957

<sup>2</sup>Sekolah Teknik Elektro dan Informatika (STEI) ITB, Jalan Ganesha 10 Bandung, 40132

e-mail: \*<sup>1</sup>[altianogerung@gmail.com](mailto:altianogerung@gmail.com) , <sup>2</sup>[lwikarsa@unikadelasalle.ac.id](mailto:lwikarsa@unikadelasalle.ac.id) , [rinaldi-m@stei.itb.ac.id](mailto:rinaldi-m@stei.itb.ac.id)

**Abstrak**— Desain web responsif adalah praktik untuk membuat situs web disajikan dengan baik di setiap perangkat. Sebelum pengenalan desain web responsif, pengembang web harus membuat halaman web terpisah atau bahkan situs web terpisah untuk setiap perangkat keras yang berbeda. Kekurangan dari aplikasi desain web responsif serupa antara lain ketidakmampuan untuk mengelompokkan halaman web, tidak dapat membuat *breakpoints* lokal, dan mengatur elemen HTML khusus untuk komponen tata letak. Oleh karena keterbatasan ini, Aplikasi Moli (*My Original Layout, Immediately*) dibangun untuk memungkinkan pengembang web pemula dalam merancang dan menerapkan tata letak web responsif dengan mudah dan cepat. Aplikasi ini menyediakan manajemen proyek *website*, halaman web, komponen tata letak global dan lokal, *breakpoints* global dan lokal, desain tata letak responsif serta menghasilkan dokumen kode HTML dan CSS dari tata letak web yang dirancang. Hasil pengujian mengungkapkan bahwa Aplikasi Moli berhasil mengelompokkan halaman web, membuat *breakpoint* lokal dan global, memilih dan menyesuaikan elemen HTML untuk komponen tata letak, dan menghasilkan tata letak responsif dalam bentuk dokumen kode HTML dan CSS. Juga, Aplikasi Moli dapat mengoptimalkan proses desain dengan mengurangi kode HTML yang berlebihan dengan tag presentasi, skrip, dan obyek/atribut yang dicampur dengan *structural markup*.

**Kata Kunci**—Generator, HML, CSS, Web Responsif, Desain.

## I. PENDAHULUAN

Sebelum diperkenalkan desain web responsif oleh Marcotte pada tahun 2010, pengembang web harus membuat halaman web terpisah atau bahkan situs web terpisah untuk setiap perangkat keras yang berbeda-beda [1]. Perubahan terjadi setelah Marcotte menunjukkan bagaimana desain web responsif dapat membuat pembangunan situs web dinamis yang mampu beradaptasi dengan ukuran layar dari perangkat keras apa pun. Desain web responsif mempermudah pengunjung web untuk mendapatkan tampilan konten web yang lebih baik berdasarkan kemampuan perangkat keras, seperti layar sentuh dan tampilan retina. Di sisi yang lain, pengembang web dapat menggunakan berbagai macam elemen HTML dan properti CSS, kueri media untuk menampilkan gaya CSS berbeda berdasarkan lebar layar perangkat keras, dan elemen seperti gambar, tipografi, dan navigasi. Teknik pengoptimalan kinerja dari desain web responsif dapat membuat web lebih ringan dan lebih cepat [2].

Saat ini sudah ada sejumlah aplikasi yang dikembangkan untuk membantu desain web responsif, seperti aplikasi Stoeva [3], Pingendo [4], dan UXPin [5]. Adapun keterbatasan dari beberapa aplikasi ini, antara lain ketidakmampuan untuk

mengelompokkan halaman web, membuat lokal *breakpoints* dan mengatur elemen HTML khusus untuk komponen tata letak.

Oleh karena itu, ada peluang untuk mengembangkan aplikasi serupa yang dapat mengatasi kekurangan yang sudah dijelaskan sebelumnya. Namun, fokus utama dari aplikasi ini adalah bagian tata letak dari desain web responsif. Aplikasi yang akan dibangun ini diberi nama Moli (*My Original Layout, Immediately*), dimana tersedia fungsionalitas untuk membuat proyek baru, halaman web, dan komponen tata letak serta mampu menambahkan komponen ke halaman web dan menghasilkan dokumen HTML dan CSS dari desain tata letak. Selain itu, pengaturan desain komponen tata letak dari aplikasi Moli mengikuti spesifikasi CSS yang dibuat oleh *World Wide Web Consortium* (W3C). Target pengguna aplikasi ini adalah pengembang web pemula yang hanya memiliki pengetahuan dasar tentang *box model modules* CSS, seperti lebar, tinggi, bantalan, dan margin [6]. Tujuan penelitian ini adalah untuk 1) mempermudah proses desain tata letak web responsif, 2) mempercepat proses implementasi tata letak web responsif yang menghasilkan pembuatan dokumen HTML dan CSS, 3) membantu mengatur pengelompokan halaman web, dan 4) mengaktifkan pembuatan lokal *breakpoints* dan komponen tata letak dengan elemen HTML khusus.

## II. TINJAUAN PUSTAKA

### A. Responsive web design

Dua dasar utama untuk membuat tata letak web responsif adalah menggunakan kueri media dari spesifikasi CSS dan memiliki fleksibilitas. Kueri media digunakan untuk mengubah nilai properti komponen tata letak pada perangkat keras yang berbeda-beda. Fleksibilitas merujuk pada penggunaan unit relatif, seperti %, Em, dan vw, untuk nilai properti [2]. Selain membantu pengunjung web, Peterson berpendapat bahwa desain web responsif ini juga dapat membantu pengembang web karena jumlah waktu yang dibutuhkan untuk membuat dan memelihara situs web responsif berkurang cukup signifikan dibandingkan dengan desain web non-responsif. Pada pembangunan web dengan menggunakan desain web non-responsif, situs dan halaman web dibuat secara terpisah untuk setiap perangkat keras yang berbeda disebabkan oleh ukuran/resolusi layar [2].

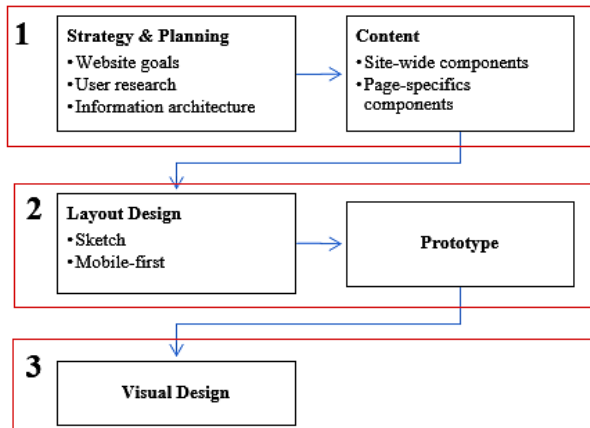
Tiga langkah dalam alur kerja desain web responsif adalah sebagai berikut [2]:

1. Identifikasi halaman web dan kontennya.

Konten dibagi menjadi 2, yaitu komponen lokal (hanya ada di satu halaman) dan global (ada di setiap halaman).

2. Membuat sketsa dan mengimplementasikan tata letak mulai dari perangkat terkecil (pengembangan *mobile-first*).
3. Membuat desain visual halaman seperti elemen warna, tipografi, dan *branding*.

Gambar 1 menggambarkan alur kerja desain web responsif.



Gambar 1. Alur Kerja Desain Web Responsif [2].

B. Penelitian Terkait

Beberapa penelitian terkait dengan pengembangan desain web responsif adalah sebagai berikut:

1. Stoeva app [3]

Aplikasi berbasis web ini digunakan untuk mengklarifikasi masalah, peluang, dan arahan yang diberikan oleh klien dengan harapan untuk menghindari kesalahpahaman dalam implementasi situs web. Aplikasi ini menyediakan komponen bawaan untuk dengan cepat mendesain halaman web. Aplikasi ini dikembangkan khusus untuk Bulgaria.

2. Pingendo [4]

Aplikasi gratis berbasis desktop ini dapat diunduh dari situs web resmi Pingendo. Aplikasi ini tersedia untuk Linux, Mac, dan Windows. Agar dapat merancang situs web responsif, aplikasi Pingendo bergantung pada kerangka *bootstrap* yang mengharuskan pengguna untuk mengetahui kerangka kerja sebelum menggunakan aplikasi.

3. UXPin [5]

Aplikasi komersial berbasis web ini memiliki fitur terbanyak dibandingkan aplikasi terkait lainnya. Beberapa fitur di antaranya adalah kolaborasi dengan orang lain, banyak pustaka komponen bawaan dan manajemen versi halaman web (juga disebut "iterasi" oleh aplikasi).

C. Teknologi Pengembangan Desain Web Responsif

Elektron, merupakan *runtime* tempat Moli berjalan, menggunakan JavaScript sebagai bahasa pemrograman. Untuk bagian *Graphical User Interface* (GUI) dari suatu program,

Elektron menggunakan HTML dan CSS. Elektron terdiri dari dua bagian utama, yaitu Node.js dan Chromium [7]. Contoh program yang dibuat menggunakan *Electron* dapat dilihat pada Gambar 2 dan kode pada Gambar 3.



Gambar 2. Contoh Program Menggunakan Elektron *Runtime*

```

var {app, BrowserWindow} = require('electron')
var window

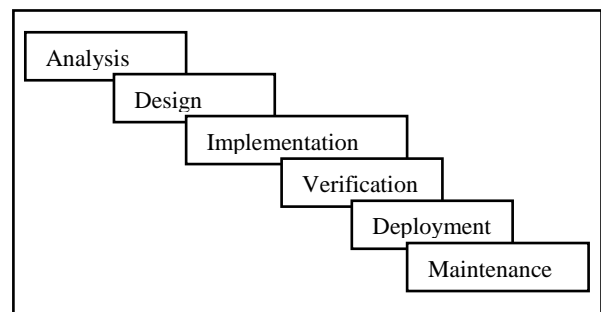
app.on('ready', () => {
  window = new BrowserWindow({
    width: 400,
    height: 400
  })

  window.loadURL(`file://${__dirname}/index.html`)
})
  
```

Gambar 3. Kode Program

III. METODE PENELITIAN

Metode Sashimi digunakan dalam pembangunan Aplikasi Moli yang mempunyai 6 tahapan, mulai dari *analyze* sampai *maintenance*. Metode ini mirip dengan *Waterfall* dengan pengecualian dimana setiap proses dalam metode Sashimi adalah saling tumpang tindih satu sama lain seperti Gambar 4 di bawah.



Gambar 4. Metode Sashimi [8]

Dengan adanya proses yang saling tumpang tindih dalam metode Sashimi, hal ini menunjukkan bahwa persyaratan tidak dapat diselesaikan sampai arsitektur setidaknya dieksplorasi sebagian, dan arsitektur tidak dapat diselesaikan sampai desain modul setidaknya dieksplorasi sebagian, dan seterusnya. Juga, kompleksitas dapat diatasi pada awal siklus hidup proyek selama fase analisis.

Dua kakas model yang digunakan dalam proses desain Moli adalah *use case diagram* dan *class diagram*. *Use case diagram* menggambarkan interaksi antara pengguna dengan fungsionalitas yang disediakan [9] dan *class diagram* untuk merancang arsitektur perangkat lunak dalam tampilan tingkat tinggi [9] [10].

#### IV. HASIL DAN PEMBAHASAN

Pembangunan Aplikasi Moli mengikuti proses-proses yang ada pada metodologi Sashimi. Sebelum proses desain dan implementasi, perlu dilakukan untuk menganalisis tiga penelitian terkait. Hasil analisis adalah spesifikasi kebutuhan untuk proses desain. Dalam proses desain, *use case diagram* dan *class diagram* digunakan untuk memodelkan aplikasi yang akan digunakan lebih lanjut dalam implementasi Moli.

##### A. Analisa Penelitian Terkait

Fitur perbandingan dari penelitian terkait adalah untuk mengidentifikasi fitur umum dan penting dari alat generator. Dalam tabel perbandingan berikut, aplikasi Stoeva dilambangkan dengan huruf S, Pingendo dengan huruf P, dan UXPin dengan huruf U.

Tabel 1.  
Perbandingan Fitur [3] [4] [5]

Fitur	Deskripsi	S	P	U
Proyek	Buat, perbarui, dan hapus proyek situs web yang responsif.			√
Halaman web	Buat, perbarui, dan hapus halaman web responsif.	√	√	√
Pembaharuan langsung dokumen HTML	Memperbaharui dokumen HTML yang ada dengan gaya (CSS) dan muat ulang otomatis jika file diubah oleh aplikasi lain.		√	
Impor tema	Menyertakan gaya dari dokumen CSS yang ada.		√	
Halaman terbaru	Tampilkan halaman web yang terakhir dibuka.		√	
Buat, perbarui, dan hapus	Gandakan atau pindahkan komponen tata letak pada kanvas desain.		√	√
Komponen bawaan	Gunakan komponen yang ada yang disediakan oleh aplikasi	√	√	√
Komponen khusus	Membuat, memperbarui, dan menghapus komponen tata letak khusus yang dibuat oleh pengguna.			√
Komponen pengelompokkan	Membuat, memperbarui, dan menghapus komponen tata letak yang terdiri dari satu atau beberapa komponen.			√
Komponen pintar	Seperti komponen khusus tetapi komponen akan memiliki pengaturan global yang sama terlepas dari halaman apa yang ditambahkan.			√
Komponen pencarian	Cari komponen bawaan atau cerdas.			√

Pesan tempel	Buat, perbarui dan hapus catatan yang bisa menempel di kanvas.			√
Komponen penataan	Gunakan komponen desain khusus oleh aplikasi untuk membuat struktur tempat komponen lain dapat ditambahkan.		√	
Desain halaman web	Menambahkan / menghapus komponen tata letak di halaman web dan juga mengubah pengaturan desain komponen.	√	√	√
Breakpoint	Buat, perbarui, dan hapus breakpoints.			√
Lapisan	Atur hierarki komponen dalam kanvas.			√
Iterasi	Buat pos pemeriksaan yang menyimpan desain halaman web pada saat itu sehingga pengguna dapat kembali ke waktu itu.			√
Grid	Layout grid untuk membantu mendesain margin antar komponen.			√
Panduan cerdas	Secara otomatis mengatur komponen dengan komponen lain jika komponen yang dipilih berada di dekat garis bantu.			√
Undo/redo	Membatalkan atau mengulang keputusan desain yang dibuat ke kanvas.	√	√	√
Zoom dan layer penuh	Memperbesar/memperkecil area kanvas dan/atau menampilkan kanvas penuh pada layar.	√		√
Kontak	Menghubungi pembuat situs web atau perwakilan untuk mendapatkan bantuan.	√	√	√
Bantuan	Teks atau <i>video tutorial</i> cara penggunaan aplikasi.			√
Generate	Membuat dokumen HTML dan CSS untuk halaman web yang sudah dirancang.	√	√	√

Selain analisis fitur dari penelitian terkait, ada juga analisis antarmuka pengguna untuk mengidentifikasi komponen *user interface* (UI) umum dari alat generator. Dengan menggunakan komponen UI yang dikenal umum, target pengguna dapat memahami bagian-bagian yang diperlukan untuk membuat jendela aplikasi yang dikembangkan.

##### B. Spesifikasi Persyaratan

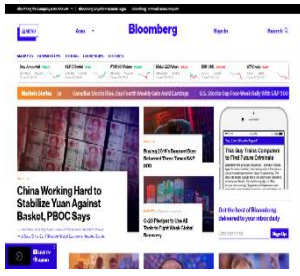
Berikut ini daftar spesifikasi persyaratan untuk pembangunan Aplikasi Moli:

1. Manajemen proyek *website*
  - a. Membuat proyek *website*.
  - b. Membuka proyek situs web yang ada secara manual, dari daftar proyek yang baru dibuka, dari daftar proyek yang disematkan.

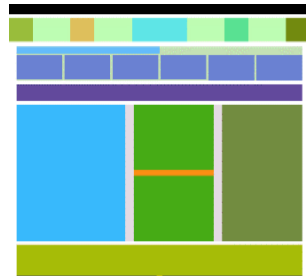








Gambar 13. Bloomberg di Desktop



Gambar 14. Hasil Tata Letak di Desktop

Berdasarkan hasil pengujian, Aplikasi Moli dapat mengoptimalkan proses desain dengan mengurangi kode HTML yang berlebihan dengan tag presentasi, skrip, dan obyek/atribut yang dicampur dengan *structural markup*. Dengan mengurangi ukuran kode HTML dan file CSS, Aplikasi Moli mampu membuat situs web memuat lebih aman dan cepat. Selain itu, aplikasi ini juga menyediakan panduan untuk pengembang web responsif pemula.

Selain mendesain situs web yang ada, ada juga pengujian pengguna untuk menemukan kekurangan dari desain antarmuka dan fungsionalitas yang disediakan dalam Aplikasi Moli. Perbaikan sudah dilakukan berdasarkan saran dari target pengguna dan kelemahan yang didapati oleh mereka saat mencoba Aplikasi Moli.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Aplikasi Moli terbukti mampu mendesain dan juga menghasilkan tata letak web yang responsif. Selain itu, Moli juga dapat:

1. Mengatur halaman web ke dalam *folder*.
2. Membuat *breakpoints* global dan lokal.
3. Membuat komponen tata letak khusus dengan elemen HTML khusus.
4. Menggandakan komponen anak.
5. Memberikan informasi panduan untuk pengembang web responsif pemula.
6. Memberikan opsi untuk menambahkan kelas CSS ke komponen tata letak.

### B. Saran

Ada beberapa rekomendasi untuk perbaikan lebih lanjut:

1. Sistem pencatatan (*logging system*) untuk merekam interaksi pengguna dengan aplikasi. Data ini dapat dikirim untuk memecahkan masalah jika ada masalah.
2. Pengujian *benchmark* untuk menganalisis kinerja aplikasi dan penggunaan memori RAM.

3. Mendukung desain web responsif penuh, seperti warna dan tipografi.
4. Mendukung pengembangan situs web dinamis menggunakan bahasa pemrograman sisi *server* seperti PHP.
5. Opsi preferensi pengguna, contoh tombol pintas dan tema aplikasi, untuk membuat pengguna merasa lebih nyaman saat menggunakan aplikasi.

## DAFTAR PUSTAKA

- [1] E. Marcote, Responsive Web Design, New York: Jeffrey Zeldman, 2011.
- [2] C. Peterson, Learning Responsive Web Design, Canada: O'Reilly Media Inc, 2014.
- [3] M. Stoeva, "Interactive Multimedia Tool for Dynamic Generation of Web Interfaces with HTML5/PHP/MySQL and JavaScript," *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 9, pp. 412-418, 2014.
- [4] Pingendo, "Pingendo - web authoring with comfort," Pingendo, n.d.. [Online]. Available: <http://pingendo.com/>. [Accessed 20th April 2020].
- [5] UXPin, "UXPin - UX Design, Wireframing Tools, Prototyping Tools," UXPin, n.d.. [Online]. Available: <http://uxpin.com>. [Accessed 21st April 2020].
- [6] World Wide Web Consortium (W3C), "W3C - CSS Basic Box Model," 2007. [Online]. Available: <https://www.w3.org/TR/css3-box/>. [Accessed 10th Maret 2020].
- [7] Electron, "Electron - Quick Start," n.d. [Online]. Available: <http://electron.atom.io/docs/v0.37.5/tutorial/quick-start/>. [Accessed 5th Desember 2019].
- [8] P. Deitel and H. Deitel, Java for Programmers 2nd Ed., Boston: Pearson, 2012.
- [9] P. Matkovic and P. Tumbas, "A Comparative Overview of the Evolution of Software Development Models," *Journal of Industrial Engineering and Management*, vol. 1, no. 4, pp. 163 - 172, 2010.
- [10] R. Stephens, Beginning Software Engineering., Indiana: John Wiley & Sons Inc, 2015.
- [11] K. Fakhroutdinov, "The Unified Modeling Language," UML-Diagrams, n.d. [Online]. Available: <http://www.uml-diagrams.org>. [Accessed 21st Januari 2020].