

Penerapan Metode Content Management System Untuk Mempermudah Mengubah Tampilan Website

Yohanes Setiawan¹, Silvia Rostianingsih², Indro Setiawan³
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131 Surabaya 60236, Indonesia
Telp(031)2983455 Fax.(031)8417658

m26414086@john.petra.ac.id¹, silvia@.petra.ac.id², indro.setiawan@peter.petra.ac.id³

ABSTRAK

Kemajuan di bidang sains dan teknologi membawa berbagai masalah dan tantangan baru. Salah satunya adalah kebutuhan untuk mengubah tampilan *website* secara cepat dan mudah. Permasalahannya masih banyak pemilik *website* yang belum mengenal *coding* sehingga untuk merubah tampilan saja cukup memakan waktu dan sulit. Di penelitian ini dilakukan percobaan dengan mengkategorikan setiap bagian *website*. setiap bagian tersebut diberi *variable* yang dapat membaca inputan dari pengguna dan memasukkannya ke dalam *database*. sistem kemudian membaca *database* tersebut dan mengubah tampilannya sesuai data yang ada di *database*. Hasil pengujian menunjukkan bahwa metode pembacaan *variable* dan *database* dapat mengubah tampilan *website*. tampilan *website* dapat diubah tanpa harus mengubah *code* didalamnya.

Kata Kunci: *Content Management System, Website, Database, Variable.*

ABSTRACT

Advances in science and technology bring new problems and challenges. One of them is the need to change the appearance of the website quickly and easily. The problem is that there are still many website owners who are not familiar with coding so changing the display is quite time consuming and difficult. In this study an experiment was conducted by categorizing each part of the website. each part is given a variable that can read input from the user and enter it into the database. the system then reads the database and changes its appearance to the data in the database. The test results show that the method of reading variables and databases can change the appearance of the website. website appearance can be changed without having to change the code in it.

Keywords: *Content Management System, Website, Database Variable.*

1. PENDAHULUAN

Pada era teknologi informasi saat ini, dibutuhkan suatu teknologi yang memungkinkan setiap orang untuk melakukan sesuatu secara mudah dan cepat. Misalnya seperti mengubah tampilan website. Untuk melakukan perubahan tampilan *website* diperlukan sedikit pemahaman tentang *coding* pada sebuah website baik itu html, php maupun css. Masalah yang mungkin dialami adalah jika terdapat pengguna yang tidak memiliki pengetahuan sama sekali dalam *coding* sebuah *website* dan ingin mengganti tampilan halaman *website* tersebut secara mandiri.

Salah satu cara agar pengguna tersebut dapat mengganti tampilan sebuah *website* sesuai dengan keinginannya adalah dengan menempatkan berbagai variabel di setiap kategori tampilan *website* sehingga tampilan *website* dapat berubah sesuai dengan nilai *variable* yang diinputkan pengguna.

Berdasarkan masalah tersebut penulis membuat sebuah sistem *website* yang dapat dipersonalisasi sehingga pengguna yang tidak memiliki pemahaman akan *coding* sebuah *website* dapat merubah tampilan *website* tersebut.

2. TINJAUAN PUSTAKA

2.1 Website

Situs web adalah kumpulan sumber daya web terkait, seperti halaman web, konten multimedia, yang biasanya diidentifikasi dengan nama domain umum, dan diterbitkan pada setidaknya satu server web. Contoh penting adalah wikipedia.org, google.com, dan amazon.com [1]. Situs web dapat diakses melalui jaringan Protokol Internet (IP) publik, seperti internet, atau jaringan area lokal pribadi (LAN), dengan merujuk pada pencari sumber daya seragam (URL) yang mengidentifikasi situs.

Situs web dapat memiliki banyak fungsi dan dapat digunakan dalam berbagai mode; situs web dapat berupa situs web pribadi, situs web perusahaan untuk perusahaan, situs web pemerintah, situs web organisasi, dll. Situs web biasanya didedikasikan untuk topik atau tujuan tertentu, mulai dari hiburan dan jejaring sosial hingga menyediakan berita dan pendidikan. Semua situs web yang dapat diakses publik secara kolektif membentuk *World Wide Web*, sementara situs web pribadi, seperti situs web perusahaan untuk karyawannya, biasanya merupakan bagian dari intranet.

Halaman web, yang merupakan blok bangunan situs web, adalah dokumen, biasanya disusun dalam teks biasa yang diselengi dengan petunjuk format Bahasa *Hypertext Markup Language* (HTML, XHTML). Mereka dapat menggabungkan elemen dari situs web lain dengan jangkar markup yang sesuai. Halaman web diakses dan diangkut dengan *Hypertext Transfer Protocol* (HTTP), yang secara opsional dapat menggunakan enkripsi (HTTP Secure, HTTPS) untuk memberikan keamanan dan privasi bagi pengguna. Aplikasi pengguna, seringkali *browser* web, membuat konten halaman sesuai dengan instruksi *markup* HTML ke terminal tampilan.

Hyperlinking antara halaman web menyampaikan kepada pembaca struktur situs dan memandu navigasi situs, yang sering dimulai dengan halaman rumah yang berisi direktori konten situs web. Beberapa situs web memerlukan pendaftaran atau

berlangganan pengguna untuk mengakses konten. Contoh situs web berlangganan mencakup banyak situs bisnis, situs web berita, situs web jurnal akademik, situs web *game*, situs web berbagi *file*, papan pesan, *email* berbasis web, situs web jejaring sosial, situs web yang menyediakan data pasar saham waktu nyata, serta situs web yang menyediakan berbagai layanan lainnya [4]. Pengguna akhir dapat mengakses situs web di berbagai perangkat, termasuk komputer desktop dan laptop, komputer tablet, ponsel cerdas, dan TV pintar.

2.2 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) adalah bahasa lembar gaya yang digunakan untuk menggambarkan penyajian dokumen yang ditulis dalam bahasa markup seperti HTML. CSS adalah teknologi landasan World Wide Web, bersama HTML dan JavaScript.

CSS dirancang untuk memungkinkan pemisahan presentasi dan konten, termasuk tata letak, warna, dan *font*. Pemisahan ini dapat meningkatkan aksesibilitas konten, memberikan lebih banyak fleksibilitas dan kontrol dalam spesifikasi karakteristik presentasi, memungkinkan beberapa halaman web untuk berbagi format dengan menentukan CSS yang relevan dalam *file .css* terpisah, dan mengurangi kompleksitas dan pengulangan dalam konten struktural [2].

Pemisahan pemformatan dan konten juga memungkinkan untuk menyajikan halaman *markup* yang sama dalam gaya yang berbeda untuk metode rendering yang berbeda, seperti pada layar, dalam cetakan, dengan suara (melalui *browser* berbasis ucapan atau pembaca layar), dan pada berbasis *Braille* perangkat taktil. CSS juga memiliki aturan untuk pemformatan alternatif jika konten diakses pada perangkat *seluler*.

Nama cascading berasal dari skema prioritas yang ditentukan untuk menentukan aturan gaya mana yang berlaku jika lebih dari satu aturan cocok dengan elemen tertentu. Skema prioritas berjenjang ini dapat diprediksi. Spesifikasi CSS dikelola oleh *World Wide Web Consortium* (W3C). Jenis media internet (tipe MIME) teks / css terdaftar untuk digunakan dengan CSS oleh RFC 2318 (Maret 1998). W3C mengoperasikan layanan validasi CSS gratis untuk dokumen CSS. Selain HTML, bahasa markup lainnya mendukung penggunaan CSS termasuk XHTML, XML polos, SVG, dan XUL.

2.3 Bootstrap

Bootstrap adalah framework front-end gratis dan open-source (perpustakaan) untuk mendesain situs web dan aplikasi web. Ini berisi template desain berbasis HTML dan CSS untuk tipografi, bentuk, tombol, navigasi dan komponen antarmuka lainnya, serta ekstensi JavaScript opsional. Tidak seperti banyak kerangka web, itu hanya menyangkut pengembangan *front-end*. Bootstrap adalah proyek paling berbintang kedua di GitHub, dengan lebih dari 126.000 bintang.

2.4 JavaScript

JavaScript sering disingkat JS, adalah bahasa pemrograman tingkat tinggi yang diinterpretasikan. Ini adalah bahasa yang juga dicirikan sebagai dinamis, lemah diketik, prototipe berbasis dan multi-paradigma. Di samping HTML dan CSS, JavaScript adalah salah satu dari tiga teknologi inti dari *World Wide Web*. JavaScript memungkinkan halaman web interaktif dan karenanya merupakan bagian penting dari aplikasi web. Sebagian besar situs

web menggunakannya, dan semua *browser* web utama memiliki mesin JavaScript khusus untuk menjalankannya.

Sebagai bahasa multi-paradigma, JavaScript mendukung gaya pemrograman yang digerakkan oleh peristiwa, fungsional, dan imperatif (termasuk berorientasi objek dan prototipe). Ini memiliki API untuk bekerja dengan teks, array, tanggal, ekspresi reguler, dan DOM, tetapi bahasa itu sendiri tidak termasuk I / O, seperti jaringan, penyimpanan, atau fasilitas grafik, mengandalkan ini pada lingkungan *host* di mana itu tertanam.

Awalnya hanya menerapkan sisi klien di browser web, mesin JavaScript sekarang tertanam di banyak jenis perangkat lunak host lainnya, termasuk sisi server di server web dan *database*, dan dalam program *non-web* seperti pengolah kata dan perangkat lunak PDF, dan dalam *runtime* lingkungan yang membuat JavaScript tersedia untuk menulis aplikasi *seluler* dan *desktop*, termasuk widget *desktop*. Istilah Vanilla JavaScript dan Vanilla JS merujuk pada JavaScript yang tidak diperpanjang oleh kerangka kerja apa pun atau pustaka tambahan. Skrip yang ditulis dalam Vanilla JS adalah kode JavaScript biasa. Meskipun ada kesamaan antara JavaScript dan Java, termasuk nama bahasa, sintaksis, dan pustaka standar masing-masing, kedua bahasa tersebut berbeda dan sangat berbeda dalam desain; JavaScript dipengaruhi oleh bahasa pemrograman seperti *Self* dan *Scheme*.

2.5 Content Management System (CMS)

Sistem manajemen konten (Inggris: *content management system*, disingkat CMS), adalah perangkat lunak yang digunakan untuk menambahkan atau memanipulasi (mengubah) isi dari suatu situs web. Umumnya, sebuah CMS (Content Management System) terdiri dari dua elemen:

- Aplikasi manajemen isi (*Content Management Application*, CMA)
- Aplikasi pengiriman isi (*Content Delivery Application*, CDA)

Elemen CMA digunakan untuk mengelola konten yang mungkin tidak memiliki pengetahuan mengenai HTML untuk mengelola pembuatan, perubahan, dan penghapusan isi dari suatu situs web tanpa perlu memiliki keahlian sebagai seorang *webmaster* [5]. Elemen CDA menggunakan dan menghimpun informasi-informasi yang sebelumnya telah ditambah, dikurangi atau diubah oleh pemilik situs web untuk memperbaharui situs web tersebut. Kemampuan atau fitur dari sebuah sistem CMS berbeda-beda. Walaupun begitu, kebanyakan dari perangkat lunak ini memiliki fitur publikasi berbasis *Web*, manajemen *format*, kontrol revisi, pembuatan indeks, pencarian, dan pengarsipan.

CMS merupakan situs web yang menerapkan sistem ini berorientasi terhadap konten. Sudah bukan merupakan kendala yang berarti bagi manajemen atau humas suatu perusahaan/instansi/organisasi untuk memperbaharui situs webnya. Dengan hak akses dan otoritas masing-masing, setiap bagian dari perusahaan/instansi/organisasi dapat memberikan kontribusinya kedalam *website* tanpa prosedur yang sulit.

Pada umumnya sebuah CMS memiliki 2 bagian kategori yaitu bagian *Front-end* dan *Back-end* [3]. Kecanggihan dan fitur masing-masing CMS bergantung pada CMS yang digunakan.

Penggunaan sistem hierarki pengguna yang diterapkan CMS dalam hak aksesnya pun sangat bervariasi sesuai CMS masing-masing. Mulai dari level akses user anggota yang hanya dapat mengirimkan data tertentu berupa komentar, kemudian editor yang dapat mengirimkan suatu artikel/berita (untuk CMS yang menyediakan fasilitas ini), hingga level *administrator* yang dapat melakukan semua fitur yang ada.

Keberadaan aplikasi gratisan di internet dan juga komunitas sumber terbuka yang semakin menjamur ikut memberikan andil yang signifikan untuk menjadikan teknologi CMS menjadi murah dari segi harga akan tetapi dengan fitur-fitur yang semakin lengkap dan canggih. Dari segi biaya implementasi pembuatan CMS sangatlah murah apalagi jika dibandingkan dengan fitur-fitur dan kemudahan yang ditawarkan. Hal inilah yang akhirnya ikut mempopulerkan CMS dan akhirnya akan menggantikan semua website konvensional yang ada.

Salah satu perangkat lunak *Content Management System* yang dikenal luas yaitu MediaWiki, perangkat lunak yang dipakai di Wikipedia dan proyek-proyek sejenis.

3. ANALISIS DAN DESAIN

3.1 Analisis

Website ini berfokus pada bagaimana mempermudah pengguna dalam mengubah tampilan *website*-nya sehingga diberikan banyak *variable* dalam *website* tersebut. *variable-variable* tersebut mewakili setiap kategori dalam *website* yang dapat diubah. *Website* yang digunakan adalah *website* dengan kategori tampilan meliputi: *header, navigation, article, side, dan footer* seperti pada *website* pada umumnya seperti *jawapos.com, tribunnews.com, dan kompas.com*

Website ini menggunakan Bahasa pemrograman php dan platform html5. Menggunakan aplikasi xampp untuk server apache dan database mysql. Pengelolaan database menggunakan php myadmin. Dan untuk hosting *website* menggunakan local host

Website ini terhubung dengan database untuk mengetahui keinginan pengguna. Sehingga pengguna harus login terlebih dahulu untuk menggunakan *website* ini. Database ini memiliki tabel-tabel untuk menyimpan *background-color, background-image, font-color, font-size, font-type, font-effect, letak side, lebar side, dan isi* setiap kategori *website* kesukaan setiap pengguna.

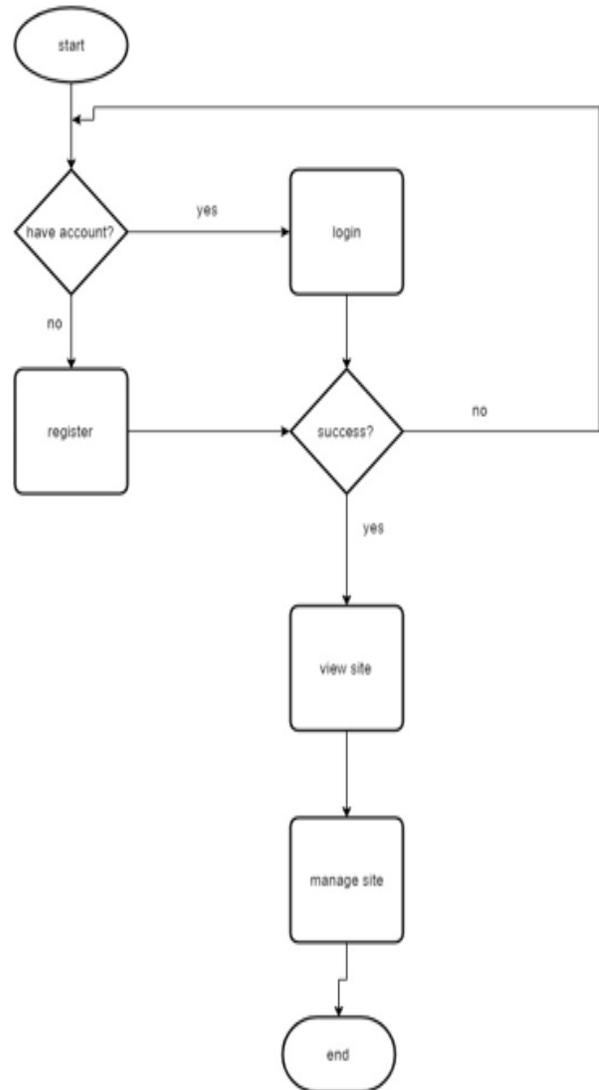
Pengguna yang tidak memiliki akun tidak dapat *login*. Pengguna dapat melakukan registrasi untuk mendapatkan akun. Diawal, saat pengguna registrasi diberikan tampilan default oleh *website* ini. Sehingga pada saat masuk ke dalam *website* ditampilkan tampilan *default*. Sehingga setiap pengguna baru memiliki tampilan yang sama.

Untuk mengubah tampilan *default* tersebut pengguna dapat pergi ke halaman *manage*. Di halaman *manage* pengguna diberikan *choice* untuk setiap kategori tampilan. Setelah pengguna mengklik *save* maka *website* ini berubah sesuai *choice* yang dipilih pengguna tersebut. Jika pengguna belum puas dengan perubahan

yang dipilihnya atau merasa ada yang kurang, pengguna dapat *manage* lagi berulang-ulang hingga mendapat tampilan yang sesuai keinginan pengguna.

3.2 Desain

Website ini menggunakan *variable* untuk mendeteksi tampilan *website* yang diinginkan pengguna. Berikut adalah alur dari *website* ini seperti pada Gambar 1.



Gambar 1. Flowchart System

Penjelasan Flowchart System:

Pertama-tama aplikasi akan mengecek apakah pengguna merupakan pengguna lama yang telah memiliki akun atau merupakan pengguna baru yang belum mempunyai akun. Jika pengguna merupakan pengguna baru, pengguna dapat mendaftar untuk memiliki akun. Jika pendaftaran berhasil maka pengguna langsung dapat masuk ke dalam aplikasi. Jika gagal, pengguna dapat melakukan pendaftaran ulang hingga berhasil.

Untuk pengguna lama, pengguna dapat langsung *log in* tanpa harus registrasi kembali. Jika *log in* berhasil pengguna dapat masuk ke dalam aplikasi. Jika gagal pengguna harus *log in* kembali hingga berhasil. Setelah berhasil masuk ke dalam aplikasi pengguna bebas merubah tampilan *website* sesuai keinginannya

4. IMPLEMENTASI

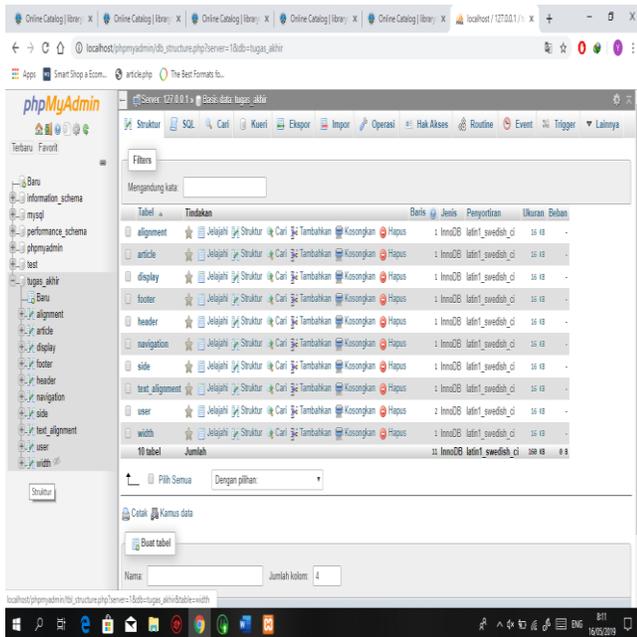
Aplikasi ini diimplementasikan seperti wordpress. Aplikasi ini menggunakan *hypertext mark-up language* (html) untuk *user interface* (UI) nya. Aplikasi ini juga menggunakan *hypertext preprocessor* (PHP) untuk system dalam aplikasi, serta menggunakan *database* mysql phpMyAdmin untuk database para pengguna. Penggunaan jaringan lokal untuk hosting (*local host*), menggunakan *server* apache serta XAMPP sebagai *third application*.

5. HASIL PENGUJIAN

Secara garis besar pengujian dikelompokkan menjadi empat kelompok besar. Berikut adalah dua kelompok besar pengujian yaitu pengujian *database* dan pengujian fungsi.

5.1 Pengujian Database

Pengujian ini dilakukan untuk memastikan apakah *database* sudah berjalan baik dan benar serta apakah isi di setiap *table* di *database* sesuai. Pengujian database berupa *register*, *login*, dan *connection*. *Database* dapat dilihat di Gambar 2.



Gambar 2. Database

5.2 Pengujian Fungsi

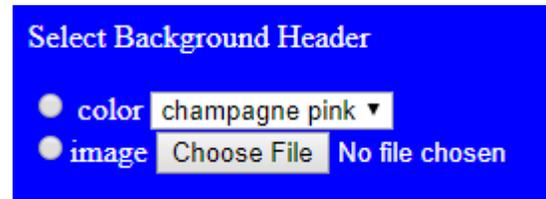
Pengujian ini dilakukan untuk memastikan apakah fungsi-fungsi di website sudah berjalan baik dan benar. Pengujian fungsi meliputi pergantian *background* (*color and or image*), *font* (*size, type, and or color*), tata letak *side*, dan jumlah artikel yang ingin ditampilkan. Berikut adalah contoh dari beberapa pengujian yang dilakukan.

5.2.1 Pergantian Background Header

Pengujian ini memastikan apakah *background header* benar-benar terganti sesuai keinginan pengguna. *Default background header* dapat dilihat di Gambar 3. Proses pemilihan gambar oleh pengguna dapat dilihat di Gambar 4 sementara hasil dapat dilihat di Gambar 5



Gambar 3. Default Background Header



Gambar 4. Pengguna Memilih Background Header



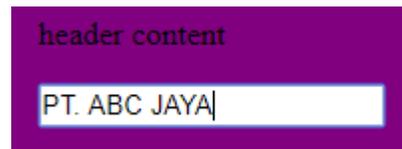
Gambar 5. Background Header Terganti

5.2.2 Pergantian Isi text Header

Pengujian ini dilakukan untuk memastikan apakah isi *text header* benar-benar terganti sesuai dengan *text* yang ditulis pengguna. *Default text header* dapat dilihat Gambar 6. Proses pergantian *text* oleh pengguna dapat dilihat di Gambar 7 sementara hasilnya dapat dilihat di Gambar 8.



Gambar 6. Default Text Header



Gambar 7. Pengguna Merubah Text Header



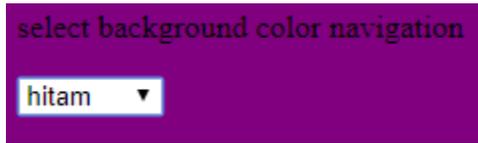
Gambar 8. Text Header Berubah

5.2.3 Pergantian Background Navigation

Pengujian ini dilakukan untuk memastikan apakah *background color navigation* benar-benar terganti sesuai dengan yang dipilih pengguna. *Default text header* dapat dilihat di Gambar 9. Proses pergantian *background color* oleh pengguna dapat dilihat di Gambar 10 sementara hasilnya dapat dilihat di Gambar 11.



Gambar 9. Default Background Color Navigation



Gambar 10. Pengguna Memilih Background Color Navigation



Gambar 11. Background Color Navigation Berubah

5.2.4 Pergantian Isi Text Navigation

Pengujian ini dilakukan untuk memastikan apakah *text navigation* benar-benar terganti sesuai dengan yang dipilih pengguna. *Default text navigation* dapat dilihat di Gambar 12. Proses pergantian *text* dapat dilihat di Gambar 13 sementara hasilnya dapat dilihat di Gambar 14.



Gambar 12. Default Text Navigation



Gambar 13. Pengguna Mengganti Text Navigation



Gambar 14. Text Navigation Berubah

5.2.5 Pergantian Isi Text Artikel

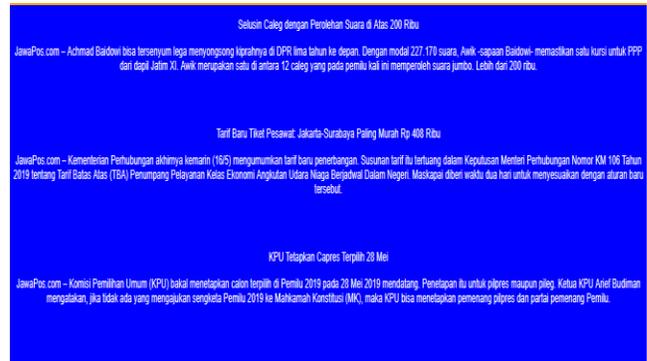
Pengujian ini dilakukan untuk memastikan isi *text article* berubah sesuai dengan keinginan pengguna. *Default text* artikel dapat dilihat di Gambar 15. Proses pergantian *text* oleh pengguna dapat dilihat di Gambar 16 sementara hasilnya dapat dilihat di Gambar 17.



Gambar 15 Default Text Article



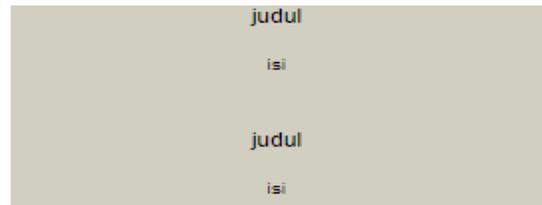
Gambar 16 Pergantian Kata oleh Pengguna



Gambar 17 Isi Text Artikel Terganti

5.2.6 Pergantian Font Side

Pengujian ini dilakukan untuk memastikan *font side* berubah sesuai dengan keinginan pengguna. *Default font side* dapat dilihat di Gambar 18. Proses pergantian *font* oleh pengguna dapat dilihat di Gambar 19 sementara hasilnya dapat dilihat di Gambar 20.



Gambar 18 Default Font Side



Gambar 19. Pemilihan Font Oleh Pengguna



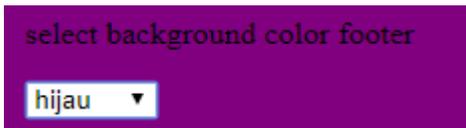
Gambar 20. Font Side Berubah

5.2.7 Pergantian Background Footer

Pengujian ini dilakukan untuk memastikan *background footer* berubah sesuai dengan keinginan pengguna. *Default background footer* dapat dilihat di Gambar 21. Proses pergantian *background* oleh pengguna dapat dilihat di Gambar 22 sementara hasilnya dapat dilihat di Gambar 23



Gambar 21. Default Background Footer



Gambar 22. Pemilihan Background oleh Pengguna



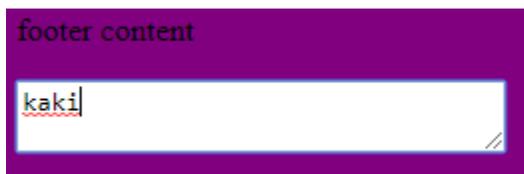
Gambar 23. Background Footer Berubah

5.2.8 Pergantian Isi Text Footer

Pengujian ini dilakukan untuk memastikan isi *text footer* berubah sesuai dengan keinginan pengguna. *Default isi text footer* dapat dilihat di Gambar 24. Proses pergantian *font* oleh pengguna dapat dilihat di Gambar 25 sementara hasilnya dapat dilihat di Gambar 26



Gambar 24. Default Isi Text Footer



Gambar 25. Pergantian Isi Text Footer Oleh Pengguna



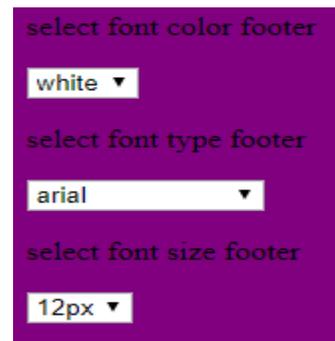
Gambar 26. Isi Text Footer Berubah

5.2.9 Pergantian Font Footer

Pengujian ini dilakukan untuk memastikan *font footer* berubah sesuai dengan keinginan pengguna. *Default font footer* dapat dilihat di Gambar 27. Proses pergantian *font* oleh pengguna dapat dilihat di Gambar 28 sementara hasilnya dapat dilihat di Gambar 29



Gambar 27. Default Font Footer



Gambar 28. Pemilihan Font Footer Oleh Pengguna



Gambar 29. Font Footer Berubah

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan berbagai hal berikut:

- Aplikasi dapat berfungsi dengan baik sesuai dengan system yang seharusnya:

- o Setiap *variable* dapat membaca inputan pengguna dan memasukkannya ke dalam database.
- o *Database* dapat menyimpan inputan pengguna
- o *Website* dapat membaca database yang berisikan inputan pengguna
- o Tampilan website berubah sesuai inputan pengguna

- Aplikasi berhasil merubah tampilan *website* tanpa harus mengubah *code* php nya

- Aplikasi berhasil membuktikan orang yang tidak mengerti *code website* dapat merubah tampilan *website* sesuai pilihannya.

6.2 Saran

Saran yang dapat diberikan untuk aplikasi ini adalah:

- Pengembangan kategori-kategori *website* yang dapat diubah
- Penambahan *platform*
- Tampilan aplikasi agar lebih menarik

7. DAFTAR PUSTAKA

- [1] Alatas, H. 2013. *Responsive Web Design Dengan PHP dan Bootstrap*. Yogyakarta: lokomedia.
- [2] KevinDekemele, AmélieChevalier and MiaLoccufier. 2018. *ODYSC: A Responsive Educational Web App For Dynamics And Control*. IFAC-PapersOnLine Volume 51, Issue 4, 2018, Pages 310-315.
- [3] Meltem, HuriBaturay and MuratBirtane. 2013. *Responsive Web Design: A New Type of Design for Web-based Instructional Content*. Procedia - Social and Behavioral Sciences Volume 106, 10 December 2013, Pages 2275-2279.
- [4] Mundzir, M.F. 2014. *Php Tutorial Book For Beginner*. Yogyakarta: Notebook.
- [5] Yanto R, M.K. 2016. *Manajemen Basis Data Menggunakan Mysql*. Yogyakarta: Deepublish