

Literature Review dan Survey Trend Teknologi Pengembangan Website untuk Website Skala Kecil

Danny Sebastian^{1,*}, Irwan Sembiring², Eko Sedyono², Kristoko Dwi Hartomo²

¹ Informatika; Universitas Kristen Duta Wacana; Jalan Dr Wahidin Sudirohusodo No 5-25 Yogyakarta 55224, (0274)563929; e-mail: danny.sebastian@staff.ukdw.ac.id.

² Doktor Ilmu Komputer; Universitas Kristen Satya Wacana; Jln Dr. O. Notohamidjojo No 1-10 Blotongan, Salatiga 50715, (0298)321212; e-mail: irwan@uksw.edu, eko@uksw.edu, kristoko@uksw.edu,

* Korespondensi: e-mail: danny.sebastian@staff.ukdw.ac.id

Diterima: 18 Januari 2023; Review: 04 Mei 2023; Disetujui: 15 Mei 2023

Cara sitasi: Sebastian D, Sembiring I, Sedyono E, Hartomo KD. 2023. *Literature Review dan Survey Trend Teknologi Pengembangan Website untuk Website Skala Kecil*. Information System for Educators and Professionals. Vol 7(2): 185-194.

Abstrak: Perkembangan teknologi *website* dari web 1.0 sampai dengan web 4.0 membuat banyak teknologi pengembangan *website*. Mulai dari Bahasa pemrograman, *Frameworks*, *Content Management System (CMS)*, *Static Site Generator (SSG)*, dan teknologi lain. Masing-masing teknologi pengembangan *website* memiliki karakteristik, kelebihan, dan kekurangannya masing-masing. Penelitian ini bertujuan untuk melihat karakteristik teknologi SSG dan CMS, kemudian menentukan teknologi man yang lebih cocok digunakan untuk *website* dengan skala kecil. Metode studi literatur dan survey digunakan untuk melihat karakteristik masing-masing teknologi. Ada 10 kriteria yang digunakan sebagai pembanding CMS dan SSG, yaitu komponen, jenis *website*, kecepatan layanan, fleksibilitas, *security*, *source-control*, *development speed* vs skala *website*, konten dinamis, *admin page*, dan *hosting*. Pendekatan SSG cocok untuk pengembangan aplikasi *website* dengan skala kecil atau *website* statis dengan sedikit interaksi dari pengguna. Pendekatan CMS cocok untuk pengembangan *website* skala menengah atau *website* dinamis dengan banyak interaksi dari pengguna. Berdasarkan hasil survey ke *website freelance* dan *website data statistik*, trend SSG masih kalah dibandingkan dengan CMS.

Kata kunci: static site generator; content management system; pengembangan *website*

Abstract: The development of websites from web 1.0 to web 4.0 has created a lot of website development technologies. Starting from programming languages, frameworks, content management systems (CMS), static site generators (SSG), and other technologies. Each website development technology has its own characteristics, advantages, and disadvantages. This study aims to analyze the differences between 2 website development technologies that are widely used today, namely using CMS and using SSG. There are 10 criteria used as a comparison between CMS and SSG, namely components, website type, service speed, flexibility, security, source-control, development speed vs website scale, dynamic content, admin page, and hosting. The SSG approach is suitable for developing small-scale website applications or static websites with little user interaction. The CMS approach is suitable for developing medium-scale websites or dynamic websites with a lot of user interaction. Based on survey results to upworks and freelancer websites, the trend of SSG is still inferior to that of CMS.

Keywords: static site generator; content management system; website engineering

1. Pendahuluan

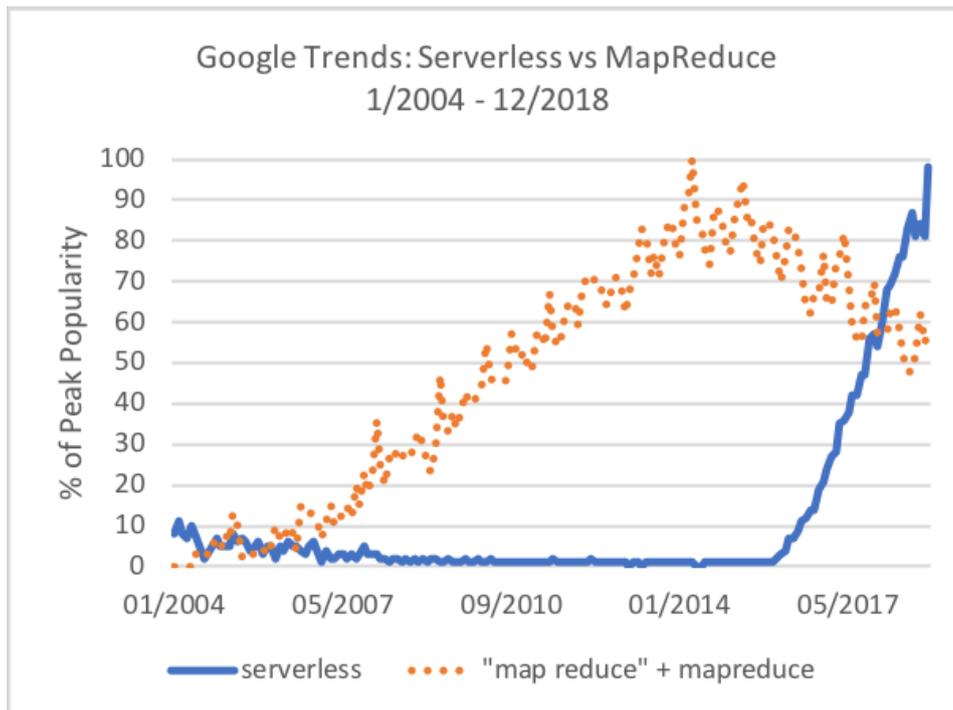
Perkembangan *website* dimulai dari web 1.0 sampai dengan web 4.0 [1] [2]. Web 1.0 adalah generasi pertama dari aplikasi *website* yang dibangun oleh Berners-Lee. Pada prinsipnya,

web.1.0 merupakan read-only *website*. Web 2.0 merupakan generasi kedua aplikasi web, dengan fitur yang menjadi fokus adalah read-write web. Web 2.0 memungkinkan adanya masukan dari pengguna dan memungkinkan adanya interaksi sosial. Web 3.0 atau semantic web merupakan pengembangan selanjutnya yang memiliki fokus mengurangi pekerjaan manusia dan memungkinkan mesin membaca konten dari *website*. Web 4.0 merupakan read-write-execution-concurrency web dengan interaksi cerdas antara manusia dengan robot. Perkembangan aplikasi *website* menjadi web 4.0 juga membuat banyak teknologi *website* baru bermunculan.

Secara umum, frameworks merupakan komponen dan pola yang dapat digunakan berulang kali pada aplikasi yang berbeda. Pengembangan aplikasi *website* menggunakan frameworks menjadi trend karena adanya beberapa manfaat [3]. Pertama, pengembangan aplikasi dapat menjadi lebih optimal dan lebih cepat karena sudah ada kerangka kerja yang dapat digunakan. Kedua, frameworks yang banyak digunakan akan memiliki *security* yang lebih baik. Ketiga, banyak frameworks yang tersedia secara gratis atau open source. Keempat, pada umumnya frameworks sudah menyediakan dokumentasi dan support yang memudahkan proses development. Pengembangan aplikasi *website* menggunakan frameworks cocok untuk membuat *website* dinamis dengan skala yang besar [4]. Dalam pengembangan aplikasi *website* menggunakan frameworks perlu mengatur beberapa hal, seperti pengaturan routes, parameter awal, dan lain sebagainya. Hal ini lebih memakan waktu dibandingkan pengembangan aplikasi *website* tanpa menggunakan frameworks. Akan tetapi setelah pengaturan awal frameworks tersebut, proses development menjadi lebih cepat. Frameworks yang digunakan untuk pengembangan aplikasi *website*, biasanya bergantung pada bahasa pemrograman yang digunakan [5]. Misal, frameworks pada bahasa pemrograman PHP adalah CakePHP2, CodeIgniter, Symfony2, Yii2, Laravel, dan lain sebagainya [3] [6]. Sedangkan pada frameworks bahasa Javascript adalah Angular JS, React JS, dan Vue JS [7] [8] [9].

Ditengah perkembangan aplikasi *website* dengan skala besar dan dinamis, pengembangan aplikasi *website* dengan skala kecil atau *website* statis kurang mendapatkan perhatian. Contoh aplikasi *website* skala kecil adalah company profile, portfolio, *website* dokumentasi, dan lain-sebagainya. Pengembangan aplikasi *website* dengan frameworks tidak cocok untuk aplikasi *website* dengan skala yang kecil. Hal ini dikarenakan biaya yang dibutuhkan untuk mempublikasikan aplikasi dengan frameworks relatif mahal, seperti biaya hosting, biaya developer, dan lain sebagainya. Aplikasi *website* dengan skala yang kecil biasanya tidak membutuhkan fitur dinamis karena konten data yang ada didalamnya relatif jarang berubah. Salah satu kelemahan aplikasi *website* statis adalah ketika ada sebuah perubahan konten, maka diperlukan pengetahuan programming untuk dapat melakukan perubahan.

Content Management System atau CMS merupakan teknologi pengembangan aplikasi *website* bagi pengguna yang tidak memiliki pengetahuan mengenai pemrograman [10]. CMS cocok digunakan untuk pemula dalam *website* development atau untuk perusahaan kecil karena *website* dapat dibuat dengan biaya yang relatif murah. CMS pengembangan *website* yang banyak digunakan adalah WordPress, Joomla dan Drupal [11]. Komponen dari CMS adalah public-facing *website* dan admin-facing *website*.



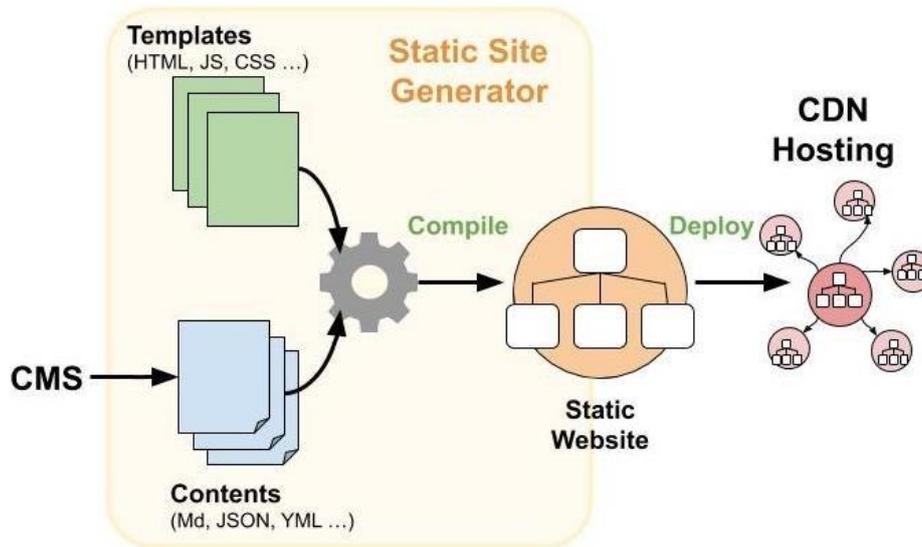
Gambar 1. Google trends Serverless dan Map Reduce tahun 2004 – 2018 [12]

Dalam beberapa tahun terakhir penyedia jasa cloud computing menawarkan programming interfaces dibawah keterangan serverless computing. 0 merupakan gambar trend serverless computing yang berkembang dalam beberapa tahun terakhir. Serverless computing merupakan layanan yang memungkinkan developer melakukan hosting aplikasi *website* pada serverless environment [13]. *Serverless Computing* sering disebut dengan *Function as a Services* atau FaaS [12]. Teknologi *serverless computing* memiliki keunggulan dibandingkan pendekatan pengembangan *website* tradisional, yaitu lebih mudah dalam melakukan pengaturan [14]. Pada FaaS, aplikasi diletakkan di komputer server dan berfungsi menggunakan fungsi-fungsi yang disediakan oleh cloud providers. Ide dari FaaS adalah simpel dan out of programming text book [12]. Pada pendekatan tradisional, developer membuat program berdasarkan fungsi-fungsi yang disesuaikan oleh input dan output, kemudian program terdiri dari fungsi-fungsi tersebut. Pada FaaS. Programming hanya perlu membuat sebuah fungsi dan mendaftarkan fungsi tersebut pada cloud, kemudian menggunakan fungsi-fungsi tersebut pada sebuah program. Serverless computing dapat digunakan untuk hosting aplikasi berbasis internet, seperti *website*, chatbot [15], dan lain sebagainya [13].

Static Site Generator atau SSG merupakan pendekatan pengembangan aplikasi *website* secara dinamis untuk menghasilkan sebuah *website* statis [16] [17]. Perbedaan SSG dengan traditional *website* application adalah SSG menghasilkan site page pada saat data template di-generate, sedangkan traditional *website* application menghasilkan site page pada saat adanya request dari pengguna. Pada SSG, pengguna mengakses *site page* yang sudah di-generate, sehingga waktu delay response menjadi sangat singkat karena tidak ada proses akses ke database dan tidak ada proses generate HTML. Dari sisi keamanan, *website* hasil dari SSG adalah *website* statis, sehingga tidak terdapat celah untuk hacker mengakses data *website*. Mengelola hosting untuk mempublikasikan *website* hasil SSG relatif lebih mudah dan murah. SSG memungkinkan penggunaan serverless computing sebagai hosting.

JAMStack merupakan salah satu teknologi pengembangan *website* yang berbasis client-side Javascript, reusable Application Programming Interface (API), dan prebuilt Markup pada sebuah Content Delivery Network (CDN) Hosting [18] [19] [20] [21]. Salah satu komponen dari JAMstack adalah SSG. Paradigma Jamstack building block dapat dilihat pada Gambar 2 JAMstack *website* menghasilkan halaman *website* menggunakan konsep SSG [21]. Contoh SSG yang tersedia pada JAMstack adalah Hugo, Gatsby, Next.js React, dan lain sebagainya [19]. Masing-masing SSG memiliki kegunaannya dan tujuannya sendiri, seperti Hugo SSG

digunakan untuk membuat simple static *website*, Gatsby atau Next.js React untuk membuat *website* yang lebih kompleks.



Gambar 2. Paradigma Jamstack *website building block* [13]

Pada pengembangan *website*, ketersediaan dokumentasi terkait teknologi yang digunakan untuk pengembangan aplikasi adalah hal yang penting. Selain tersedianya dokumentasi, ketersediaan sumber daya, tutorial, dan forum diskusi menjadi pendukung dalam memilih teknologi yang akan digunakan untuk pengembangan aplikasi *website*. Dengan memilih teknologi yang populer dan memiliki komunitas yang besar, pengembang *website* dapat memiliki lebih banyak kesempatan untuk bergabung dengan forum atau grup diskusi, memperoleh dukungan teknis, dan mendapatkan sumber daya dan tutorial yang dibutuhkan dalam pengembangan *website*. Hal ini dapat membantu mengurangi waktu dan biaya pengembangan, dan juga dapat memastikan bahwa teknologi yang dipilih dapat berkembang dan bertahan dalam jangka waktu yang panjang. Akan tetapi saat ini teknologi berkembang dengan lebih cepat, banyak teknologi baru yang muncul. Sehingga pengembang aplikasi *website* perlu lebih teliti dalam melihat jumlah pengguna dan ketersediaan sumber daya dan tutorial.

Berdasarkan permasalahan yang sudah dipaparkan, penelitian ini akan melihat perbandingan pendekatan teknologi CMS dan SSG untuk pengembangan aplikasi *website* dengan skala kecil. Analisis perbandingan dilakukan menggunakan studi literatur berdasarkan karakteristik dan pendekatan teknis dari CMS dan SSG (1), dan melihat trend penggunaan masing-masing teknologi (2). Penelitian ini akan memberikan gambaran perbandingan CMS dan SSG berdasarkan literatur yang ada dan memberikan gambaran kondisi trend penggunaan CMS dan SSG.

2. Metode Penelitian

Perbandingan antara CMS dan SSG dilakukan berdasarkan beberapa kriteria, yaitu: kecepatan dalam melayani request, fleksibilitas dalam penambahan komponen baru, *security*, *source code control*, *development speed* vs skala *website*, konten dinamis, *admin page*, dan *hosting* [22] [23]. Kriteria perbandingan didapatkan dari beberapa artikel terdahulu yang melakukan perbandingan beberapa tools pengembangan dan CMS. Data yang digunakan pada analisis ini adalah data kualitatif. Data tersebut didapatkan dari studi pustaka dan observasi mengenai karakteristik, keunggulan, dan kelemahan dari CMS dan SSG. Data kualitatif tersebut dianalisis dan ditampilkan kedalam bentuk sebuah tabel perbandingan.

Analisis kedua dilakukan untuk melihat trend SSG dan CMS dari sisi ketersediaan pekerjaan atau proyek. Jumlah ketersediaan pekerjaan dapat dijadikan acuan jumlah pengguna dan komunitas dari pengguna masing-masing CMS dan SSG. Penulis melihat jumlah pengguna dan komunitas akan berbanding lurus dengan ketersediaan dokumentasi untuk masing-masing teknologi. Analisis ini dilakukan dengan melakukan survey ke *website* penyedia jasa

freelance/outsorce, yaitu Upwork.com, Freelancer.com, dan Fastwork.id. Penulis melakukan pencarian pekerjaan dan proyek untuk masing-masing teknologi berdasarkan kata kunci. 4 kata kunci dipilih secara spesifik berdasarkan cerminan dari masing-masing teknologi. Data yang digunakan pada analisis kedua adalah data kuantitatif berdasarkan hasil pencarian dari *website* penyedia jasa freelance/outsorce. Selain melakukan pencarian ke *website* penyedia jasa freelance/outsorce, dilakukan pengumpulan data jumlah *website* yang menggunakan teknologi SSG dan CMS dari *website* penyedia statistik Similartech.com dan Builtwith.com.

3. Hasil dan Pembahasan

Perbandingan Karakteristik CMS dan SSG

Tabel perbandingan CMS dan SSG dapat dilihat pada Tabel 1. Ada beberapa kriteria yang menjadi pembanding, yaitu komponen utama, jenis *website*, kecepatan dalam melayani request pengguna, fleksibilitas penambahan komponen baru, *security*, source code control, kecepatan dibandingkan dengan skala *website*, konten dinamis, dan halaman admin.

Tabel 1. Perbandingan CMS dan SSG

No	Kriteria	Content Management System (CMS)	Static Site Generator (SSG)
1	Komponen	Front-end <i>website</i> dan back-end admin panel	Javascript, reusable API, pre-build Markup
2	Jenis <i>website</i>	<i>Website</i> Dinamis	<i>Website</i> Statis
3	Kecepatan layanan	Lebih lambat dibanding SSG, karena berbasis pada <i>website</i> dinamis, dalam melayani request perlu mengakses ke database dan menghasilkan struktur HTML secara dinamis	Lebih cepat dibanding CMS, karena berbasis <i>website</i> statis, dalam melayani request hanya mengirimkan pre-built struktur HTML
4	Fleksibilitas	Tidak fleksibel dalam penambahan komponen baru, karena harus mengikuti prosedur dan kesediaan komponen dari CMS	Lebih fleksibel dibanding CMS
5	Security	Karena berbasis pada <i>website</i> dinamis, hacker memiliki banyak celah untuk mengakses privasi.	Karena berbasis pada <i>website</i> statis, maka hacker dan malware tidak memiliki banyak celah untuk mengakses privasi. Tidak menggunakan database dan dynamic interpreter.
6	Source control	Bergantung pada update dari penyedia CMS.	Menggunakan version control system, seperti GIT. Memungkinkan fleksibilitas lebih dibandingkan CMS.
7	Development Speed vs Skala <i>website</i>	Development Speed tidak terlalu berpengaruh terhadap skala <i>website</i> , karena berbasis <i>website</i> dinamis. Beban kerja terbagi pada setiap request.	Beban kerja adalah di awal kompilasi. Karena SSG melakukan kompilasi dan menghasilkan pre-build HTML, <i>website</i> skala besar akan memakan waktu kompilasi lebih lama.
8	Konten dinamis	Lebih baik daripada SSG karena berbasis <i>Website</i> Dinamis	SSG tidak memungkinkan adanya interaksi dinamis antara <i>website</i> dengan pengguna.
9	Back-end / Admin page	Admin page yang mudah digunakan oleh pengguna udah disediakan oleh CMS. Tidak memerlukan pengetahuan pemrograman.	Tidak memiliki Admin page, sehingga perlu API-driven CMS untuk pengguna yang tidak memahami pemrograman.
10	Hosting	Membutuhkan hosting storage yang cukup besar dan pemrosesan komputasi yang tinggi	Dapat menggunakan teknologi Content Delivery Network (CDN), bisa mendapatkan versi gratis. Atau menggunakan pendekatan serverless computing.

Secara umum, kriteria kecepatan, fleksibilitas, security, source code control dan hosting merupakan keunggulan dari SSG. Dari sisi kecepatan pada melayani request, SSG memiliki keunggulan karena halaman *website* SSG di-generate menjadi sebuah *website* statis, kemudian diletakkan di web hosting. Pengguna yang melakukan request, akan dilayani oleh web server menggunakan *website* statis yang sudah dihasilkan. Karena yang diakses adalah *website* statis, maka waktu yang dibutuhkan dalam melayani request menjadi sangat singkat. Sama halnya

dengan aspek *security*, karena yang diakses adalah *website* statis, maka *website* tidak memiliki celah keamanan yang banyak. Pada *website* dinamis, celah keamanan banyak muncul karena adanya interaksi antara aplikasi *website* dengan pengguna. Karena SSG menggunakan *website* statis, maka celah keamanan menjadi tidak terlalu banyak dibandingkan aplikasi *website* atau CMS. Pada aspek fleksibilitas fitur atau halaman dan source code control, SSG memiliki keunggulan karena pada prinsipnya SSG melakukan development halaman tersebut secara dinamis sebelum di-generate menjadi *website* statis. Hal ini berbeda dengan CMS yang memiliki batasan fitur berdasarkan batasan plugin atau batasan CMS tersebut. Hosting yang dibutuhkan oleh SSG relatif lebih ringan dibandingkan dengan kebutuhan hosting CMS.

Sedangkan komponen development speed dibandingkan dengan skala *website*, konten dinamis, dan admin page merupakan keunggulan dari CMS. Pada komponen development speed dibanding skala *website* dan admin page, CMS memiliki keunggulan lebih karena pada prinsipnya CMS tidak melakukan development, tetapi melakukan setup atau pengaturan saja. Setup dan pengaturan halaman *website* dilakukan pada admin page yang sudah disediakan oleh CMS. Sedangkan pada SSG, tidak tersedia admin page untuk melakukan perubahan konten/halaman, tetapi perubahan konten/halaman dilakukan dengan cara mengubah kode program *website*. Pada aspek konten dinamis, CMS memiliki keunggulan karena setiap request dilayani secara realtime saat request diterima. Karena dilayani secara realtime, maka data yang digunakan adalah data terakhir yang ada pada database.

Berdasarkan perbandingan ini, disimpulkan SSG cocok untuk skala *website* yang kecil atau *website* statis dimana tidak ada atau minim interaksi dengan pengguna. Selain itu pemahaman pemrograman dibutuhkan untuk dapat menggunakan SSG. Sedangkan CMS cocok digunakan untuk *website* dengan skala menengah-besar, dan client yang tidak memiliki pemahaman pemrograman.

Untuk mendukung literasi yang ditemukan, dilakukan analisis langkah pengembangan *website* menggunakan CMS dan SSG. Langkah pengembangan *website* menggunakan SSG dapat dilihat pada Gambar 3. Secara umum langkah pengembangan *website* menggunakan SSG terdiri dari 7 langkah utama. Dari ketujuh langkah tersebut, ada langkah yang membutuhkan pengetahuan khusus tentang teknologi secara spesifik. Langkah pertama, pemilihan SSG membutuhkan pengetahuan tentang stack technology yang tepat untuk kebutuhan situs *website*. Ini membutuhkan pengetahuan teknologi *website* dasar seperti HTML, CSS, dan Javascript. Selain itu, instalasi SSG pada perangkat lokal membutuhkan pengetahuan dasar tentang penggunaan terminal, git, dan npm. Selanjutnya langkah kedua pemilihan headless-cms membutuhkan pengetahuan tentang RESTful API agar dapat menggunakan API end point. Kemudian pada langkah keempat pengembangan tampilan, dibutuhkan pengetahuan tentang JavaScript, khususnya Vue.JS dan React.JS. Proses development pada langkah keenam membutuhkan pengetahuan tentang cloud dan deployment aplikasi *website*.

Gambar 3. Langkah pengembangan situs *website* menggunakan SSG

Langkah pengembangan *website* menggunakan CMS dapat dilihat pada Gambar 4. Dari 8 langkah pengembangan *website* menggunakan Wordpress CMS, tidak ada langkah yang

secara spesifik membutuhkan pengetahuan pemrograman. Dari kedelapan langkah tersebut, hanya langkah pertama yang membutuhkan pengetahuan teknis, yaitu melakukan instalasi pada hosting. Akan tetapi langkah tersebut memungkinkan dilakukan menggunakan user interface cpanel sebuah hosting. Salah satu keunggulan Wordpress CMS adalah sudah tersedia instalasi yang memudahkan pengguna awam untuk melakukan instalasi. Instalasi yang dimaksud termasuk instalasi awal, instalasi tema, dan instalasi plugin. CMS juga menyediakan kustomisasi sampai tingkat tertentu menggunakan antarmuka bergambar atau graphical user interface. Pengetahuan pemrograman dibutuhkan ketika pengguna membutuhkan kustomisasi lebih lanjut dan tidak dapat ditangani oleh graphical user interface yang disediakan oleh CMS.

Gambar 4. Langkah pengembangan situs *website* menggunakan Wordpress CMS

Berdasarkan langkah pengembangan situs *website* menggunakan SSG dan CMS, dapat disimpulkan bahwa SSG membutuhkan pengetahuan teknis/pemrograman yang lebih tinggi dibandingkan dengan CMS. Karena pada SSG konten harus dibuat sebagai file statis dengan format seperti markdown atau HTML, selain itu *website* harus dideploy secara manual menggunakan tools GIT. Sedangkan pada CMS, pengaturan *website* dapat dilakukan menggunakan antarmuka grafis. Namun, meskipun SSG membutuhkan tingkat keahlian teknis/programming yang lebih tinggi, keuntungan yang didapat dari *website* yang lebih cepat, aman, dan mudah dikembangkan.

Trend Pekerjaan CMS dan SSG

Untuk melihat trend CMS dan SSG, dilakukan survey ke *website* penyedia freelance/oursource. Situs yang dipilih adalah freelancer.com, upwork.com danm Fastwork.id. Beberapa keyword dipilih untuk melihat trend tersedianya pekerjaan. Hasil pencarian dan perhitungan jumlah pekerjaan dapat dilihat pada Tabel 2 dan Tabel 3. Dari kedua tabel terlihat jumlah pekerjaan yang ditawarkan untuk topik CMS dan topik SSG sangat berbeda jauh. Terlihat jumlah pekerjaan pada topik CMS adalah 20.677 sedangkan pada topik SSG adalah 467. Dari jumlah tersebut, terlihat topik CMS masih lebih banyak diminati dibandingkan dengan SSG. Akan tetapi pengujian lebih lanjut perlu dilakukan untuk mendapatkan informasi mengenai jumlah programmer yang dapat mengerjakan masing-masing pekerjaan. Pengumpulan data juga masih terbatas pada hasil pencarian beberapa keyword berdasarkan tag, masih belum melakukan pencarian untuk pekerjaan yang tidak mencantumkan tag namun dapat dikerjakan menggunakan pendekatan SSG.

Tabel 2. Hasil pencarian pekerjaan yang menggunakan CMS pada *website* listing pekerjaan

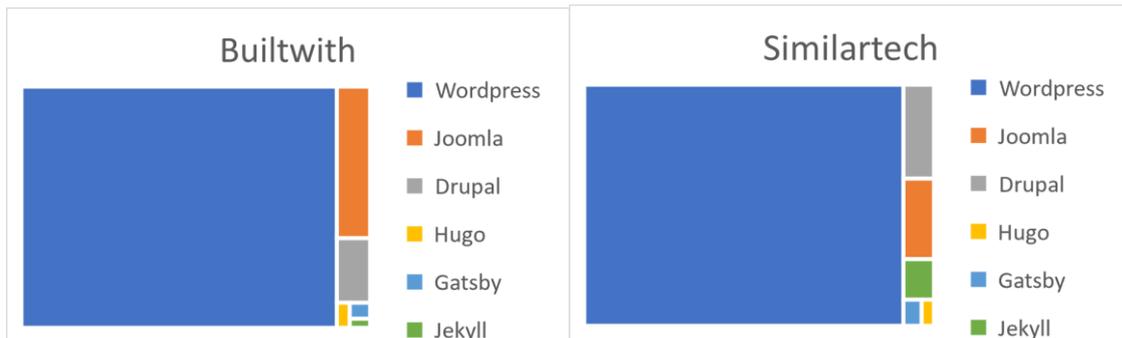
Keyword	Freelancer	Upworks	Fastwork.id
wordpress	1,241	16,052	444
joomla	30	388	4
drupal	37	291	1
cms	93	2,545	152
Jumlah	1,401	19,276	601

Tabel 3. Hasil pencarian pekerjaan menggunakan SSG pada *website* listing pekerjaan

Keyword	Freelancer	Upworks	Fastwork.id
JAMstack	1	101	152
hugo	3	20	14

ssg	0	12	14
static website	88	242	37
Jumlah	92	375	217

Analisis trend yang kedua dilakukan dengan cara melakukan survey jumlah *website* yang dikembangkan menggunakan masing-masing teknologi ke builtwith.com dan similartech.com. Hasil survey dapat dilihat pada Gambar 5. Berdasarkan data yang didapatkan, CMS wordpress, joomla dan drupal selalu berada pada jumlah pengguna pertama. Sedangkan jumlah *website* yang menggunakan teknologi SSG selalu berada pada posisi akhir. Hal ini menunjukkan trend penggunaan teknologi CMS untuk mengembangkan *website* masih lebih tinggi dibandingkan SSG.



Gambar 5. Hasil survey builtwith.com dan similartech.com

4. Kesimpulan

Beberapa kesimpulan yang dapat diambil dari studi literatur yang sudah dilakukan adalah:

- Penggunaan teknologi SSG cocok untuk pengembangan aplikasi *website* dengan skala kecil atau *website* statis. Karakteristik *website* yang sesuai adalah *website* yang tidak ada atau tidak membutuhkan banyak interaksi dengan pengguna.
- Keunggulan teknologi SSG adalah kecepatan melayani request pengguna, fleksibilitas, *security* risk rendah, source control, hosting yang ringan. Tetapi kelemahan dari teknologi SSG adalah tidak cocok untuk aplikasi *website* dengan skala yang besar, konten dinamis, dan admin page.
- Penggunaan teknologi CMS cocok untuk pengembangan aplikasi *website* dengan skala menengah atau *website* dinamis. *Website* yang membutuhkan interaksi dengan pengguna.
- Keunggulan dari teknologi CMS adalah performance pada *website* skala menengah dan besar, konten dinamis, dan admin page. Sedangkan kelemahan teknologi CMS adalah keceptan melayani request, fleksibilitas, *security* risk tinggi, source code control, dan hosting yang besar.
- Saat ini berdasarkan survey pada *website* penyedia jasa *freelance/outsorce* dan pencarian pekerjaan, Trend penggunaan teknologi CMS masih lebih banyak dibandingkan penggunaan teknologi SSG.

Saran untuk penelitian selanjutnya adalah:

- Perlu dilihat jumlah pekerjaan yang tidak mencantumkan tag terkait SSG namun dapat diselesaikan menggunakan pendekatan SSG.

Referensi

- [1] S. Aghaei, M. A. Nematbakhsh and H. K. Farsani, "Evolution of the World Wide Web: From Web 1.0 to Web 4.0," *International Journal of Web & Semantic Technology (IJWesT)*, 2012.
- [2] A. K. Ibrahim, "Evolution of the Web: from Web 1.0 to 4.0," *Qubahan Academic Journal*, vol. 1, no. 3, pp. 20-28, 2021.
- [3] N. Prokofyeva and V. Boltunova, "Analysis and practical application of PHP frameworks in development of web information systems," *Procedia Computer Science*, vol. 104, pp. 51-56, 2017.
- [4] D. Sebastian, "Pengembangan Single Website Application untuk Multiple Domain Menggunakan Laravel Frameworks," *Jurnal Sains dan Informatika*, vol. 7, no. 1, pp. 48-57, 2021.
- [5] D. H. Curie, J. Jaison, J. Yadav and J. R. Fiona, "Analysis on Web Frameworks," *Journal of Physics: Conference Series*, vol. 1362, no. 1, 2019.
- [6] M. Laaziri, K. Benmoussa, S. Khouliji, K. M. Larbi and A. E. Yamami, "A comparative study of laravel and symfony PHP frameworks," *International Journal of Electrical and Computer Engineering*, vol. 9, pp. 704-712, 2019.
- [7] E. Saks, "JavaScript Frameworks: Angular vs React vs Vue.," HAAGA-HELIA University of Applied Sciences, 2019.
- [8] P. Akther, "Angular vs. React vs. Vue Detailed Comparison Guide - Which One to Choose in 2020," 8 April 2020. [Online]. Available: <https://www.themexpert.com/blog/angular-vs-react-vs-vue>. [Accessed 15 October 2021].
- [9] P. Rawat and A. N. Mahajan's, "React JS: A Modern Web Development Framework," *International Journal of Innovative Science and Research Technology*, vol. 5, no. 11, pp. 698-702, 2020.
- [10] J.-M. Martinez-Caro, A.-J. Aledo-Hernandez, A. Guillen-Perez, R. Sanchez-Iborra and M.-D. Cano, "A comparative study of web content management systems," *Information*, vol. 9, no. 2, 2018.
- [11] V. Toma, "A Comparative Study on The Security of Web Content Management Systems," *The USV Annals of Economics and Public Administration*, vol. 21, no. 1 (33), pp. 146-151, 2021.
- [12] J. M. Hellerstein, J. Faleiro, J. E. Gonzalez, J. Schleier-Smith, V. Sreekanti, A. Tumanov and C. Wu, "Serverless computing: One step forward, two steps back," *arXiv preprint arXiv:1812.03651*, 2018.
- [13] P. Utomo and Falahah, "Building Serverless Website on GitHub Pages," in *IOP Conf. Series: Materials Science and Engineering*, 2020.
- [14] O. Paakkunainen, "Serverless computing and FaaS platform as a web application backend," Aalto University, 2019.
- [15] M. Yan, P. Castro, P. Cheng and V. Ishakian, "Building a chatbot with serverless computing," in *Proceedings of the 1st International Workshop on Mashups of Things and APIs*, 2016.
- [16] C. Diaz, "Using static site generators for scholarly publications and open educational resources," *Code4Lib Journal*, vol. 42, 2018.
- [17] A. A. Sutchenkov and A. I. Tikhonov, "Site Generator for Small Open and Private Online Courses," in *2020 International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*, Moscow, 2020.
- [18] F. Zammetti, "What Is JAMstack All About?," in *Practical JAMstack*, Springer, 2020, pp. 1-17.
- [19] D. Markovic and M. Scekcic, "Understanding Jamstack and Its Perception in Web Development," School of Innovation Design and Engineering, Sweden, 2021.
- [20] S. Peltonen, L. Mezzalana and D. Taibi, "Motivations, benefits, and issues for adopting micro-frontends: a multivocal literature review," *Information and Software Technology*, vol. 136, 2021.

- [21] D. Markovic and M. Scekcic, "Understanding Jamstack and its Perception in Web Development," Mälardalen University, Västerås, Sweden, 2021.
- [22] A. Sukarno, "Perancangan Dan Analisis Performansi Arsitektur Jamstack Studi Kasus Web Kraton Jogja," Institut Teknologi Telkom Purwokerto, 2021.
- [23] M. Laaziri, K. Benmoussa, S. Khouliji, K. M. Larbi and A. El Yamami, "A comparative study of laravel and symfony PHP frameworks," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, p. 704, 2019.
- [24] S. K. Patel, V. R. Rathod and S. Parikh, "Joomla, Drupal and WordPress-a statistical comparison of open source CMS," in *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)*, 2011.