

# Jamstack Performance and Cost-Effective Analysis with Traditional CMS Websites

Kadek Arya Putra Permana<sup>a1</sup>, I Made Agus Dwi Suarjaya<sup>a2</sup>, Ni Kadek Ayu Wirdiani<sup>a3</sup>

<sup>a</sup>Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana, Indonesia  
e-mail: <sup>1</sup>aryaputra@student.unud.ac.id, <sup>2</sup>agussuarjaya@it.unud.ac.id, <sup>3</sup>ayuwirdiani@unud.ac.id

## Abstrak

*Website menjadi sebuah media yang tidak bisa dipisahkan dari perkembangan teknologi, website dibutuhkan dikarenakan kegunaan yang praktis dan mudah, terutama sebuah website landing page yang digunakan pelaku bisnis maupun non-profit untuk menyampaikan informasi. Website dikembangkan menggunakan sebuah arsitektur yang dibangun khusus sehingga proses pengembangan dan perawatan berjalan dengan baik. Dunia berkembang semakin cepat sehingga arsitektur pengembangan website lama seiring waktu semakin sulit untuk beradaptasi, sehingga hadir arsitektur pengembangan website baru salah satunya Jamstack, arsitektur Jamstack dikatakan memiliki kelebihan dibanding arsitektur lama yang banyak digunakan hingga saat ini salah satunya adalah LAMP Stack, untuk memastikan kebenaran argumen tersebut maka diperlukan sebuah analisis yang detail sehingga mendapatkan hasil yang sesuai dengan fakta.*

**Kata Kunci:** Jamstack, LAMP Stack, Arsitektur Website, Analisis Performa, Analisis Biaya

## Abstract

*Website is a medium that cannot be separated from technological developments, websites are needed because of their practical and easy use, especially a landing page website that is used by business and non-profit people to convey information. The website is developed using a custom-built architecture so that the development and maintenance process runs well. The world is developing faster and faster so that the old website development architecture over time is increasingly difficult to adapt, so there is a new website development architecture, one of which is Jamstack, the Jamstack architecture is said to have advantages over the old architectures that are widely used until now, one of which is LAMP Stack, to ensure the correctness of these arguments, a detailed analysis is needed so as to get results that match the facts.*

**Kata Kunci:** Jamstack, LAMP Stack, Website Architecture, Performance Analysis, Cost Analysis

## 1. Introduction

Desain sebuah *website* adalah kunci dari kesuksesan sebuah *website*, untuk dapat meraih *website* yang baik diperlukan melakukan analisis *marketing*, persepsi, dan tingkah laku sehingga mendapatkan sebuah *website* yang dapat dikatakan berhasil [1]. Desain teknologi pengembangan *website* juga berpengaruh seperti teknologi yang akan digunakan. Kebutuhan yang intensif dan syarat pengembangan *website* untuk perusahaan perlu berjalan pada sistem yang cepat dan juga efisien semakin dibutuhkan [2], penggunaan arsitektur yang biasa digunakan sejak beberapa dekade sebelumnya dirasa sudah cukup lama dan diperlukan teknologi baru dalam pengembangan *website* [3].

LAMP Stack adalah arsitektur pengembangan *website* yang sudah cukup lama, dikembangkan oleh Michael Kunze pada tahun 1998, dengan struktur Linux, Apache, MySQL dan PHP serta rata-rata berbasis *open-source* sehingga dapat digunakan oleh siapa saja [4]. LAMP Stack masih tetap digunakan dikarenakan arsitekturnya yang dinamis dan cukup mudah dipahami [5]. Developer semakin membutuhkan pengembangan yang lebih efektif dari sisi kualitas, waktu pengerjaan, kebutuhan biaya, integrasi, hingga *deployment*, dikarenakan arsitektur LAMP Stack tidak efektif dalam kebutuhan tersebut maka arsitektur Jamstack dapat menjadi solusi dari masalah ini.

---

Jamstack adalah arsitektur yang relatif baru, diciptakan oleh Mathias Biilmann pada tahun 2015, Jamstack terdiri dari Javascript API dan Markup. Arsitektur Jamstack mengandalkan *client side* dalam proses *logical website* dan API sebagai *servicesnya* sehingga mengurangi beban dari *server* dan hanya terdapat *website static* pada server [6]. Jamstack arsitektur memiliki beberapa keuntungan dalam pengembangan *website* namun di samping kelebihan tersebut hadir pertanyaan bahwa apakah arsitektur Jamstack lebih baik dari arsitektur LAMP Stack, sehingga diperlukan analisis. Studi kasus pada penelitian ini menggunakan *website* yang sebelumnya sudah menggunakan arsitektur LAMP Stack kemudian dilakukan migrasi dan membandingkan beberapa poin penting yaitu dari sisi performa dan kebutuhan biaya dalam pengembangan dua arsitektur tersebut yaitu Jamstack dan LAMP Stack.

## 2. Research Method

Metode penelitian yang digunakan dalam analisis studi kasus dari arsitektur LAMP Stack dan Jamstack berupa testing dari sisi performa, *load testing* dan juga perhitungan biaya yang dibutuhkan. Migrasi diperlukan pada awal sebelum melakukan testing, *website* dilakukan migrasi dari arsitektur Jamstack yaitu menggunakan *framework* Laravel menjadi *website* dengan arsitektur Jamstack menggunakan *framework* Next.js dan API yang terhubung, *website* yang digunakan pada studi kasus adalah *website landing page* bernama Akunta.id, yaitu sebuah perusahaan *accounting*. Detail dari metode penelitian dapat dilihat pada gambar berikut.

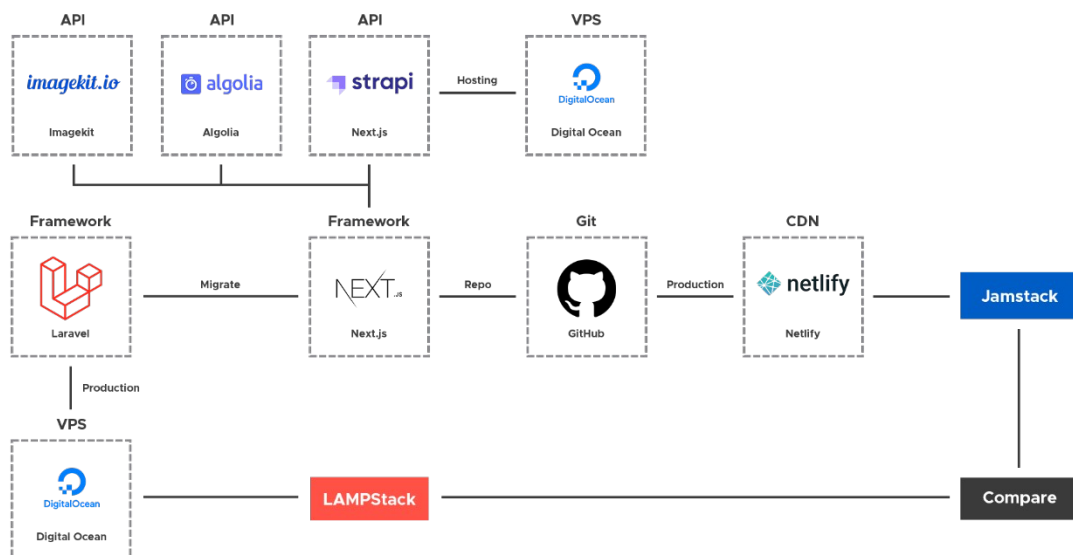


Figure 1. Struktur Penelitian LAMP Stack dan Jamstack

Error: Reference source not found menampilkan alur penelitian dua arsitektur yang dilakukan analisis, terdapat beberapa platform yang digunakan untuk mendukung dari proses alur pengembangan *website*. LAMP Stack hanya terdiri dari *framework* yang dilakukan *hosting* pada sebuah VPS dikarenakan semua *services* dan *website* berada di satu tempat, sedangkan pada Jamstack terpisah-pisah, terutama pada *services* dan *repository*. Semua alur akan berujung pada hasil akhir yang digunakan untuk melakukan komparasi dari kedua arsitektur.

## 3. Literature Study

Kajian pustaka dari penelitian analisis dua arsitektur yang berbeda menggunakan beberapa teknologi yang mendukung jalannya penelitian, beberapa teknologi tersebut akan disampaikan sebagai berikut.

### 3.1 Jamstack

Jamstack adalah singkatan dari Javascript, API dan Markup, diciptakan pertama kali oleh Mathias Biilmann pada tahun 2015 di samping itu juga Mathias adalah CEO Netlify [6]. Jamstack memiliki keamanan yang cukup baik karena tidak ada proses *backend* pada *website* hanya sebuah *static file*. Keamanan sangat penting pada sistem komunikasi [7], komunikasi yang *reliable* adalah sistem dengan level keamanan tinggi [8].

### 3.2 LAMP Stack

LAMP Stack adalah singkatan dari Linux, Apache, Markup, PHP, LAMP Stack yang diciptakan oleh Michael Kunze pada tahun 1998. LAMP Stack berfokus pada penggunaan *open-source* pada pengembangan sebuah *website* [9]. Arsitektur LAMP Stack digunakan pada pengembangan *website* dikarenakan dapat menjalankan proses dinamis pada *website*.

### 3.3 Rasio

Rasio adalah seberapa banyak nomor satu dengan lainnya dan dapat berbentuk berbagai macam seperti hitungan, nilai objek hingga jarak dan masih banyak lagi [10]. Penelitian menggunakan metode rasio dengan mencari nilai dari masing-masing arsitektur dan melihat perbedaan nilai dari dua arsitektur (1).

$$Ratio = \frac{(total\ of\ X)}{(total\ of\ Y)} \quad (1)$$

### 3.4 Web Server

*Web server* adalah *hardware* yang menerima *request* dari HTTP atau HTTPS, *end user* biasanya menggunakan *web browser* untuk melakukan komunikasi dan meminta sebuah halaman *web* menggunakan HTTP atau HTTPS, dan server memberikan respons dengan mengirimkan data yang dibutuhkan atau jika terjadi masalah akan mengirimkan pesan *error*. *Web server* juga dapat menerima *resource* dari *end user* jika dibutuhkan [11].

### 3.5 Serverless

*Serverless* adalah sebuah paradigma komputasi yang menggabungkan antara *back-end-as-a-service* (BaaS) dan *function-as-a-service* (FaaS) dengan representasi utama FaaS [12]. *Serverless* memiliki beberapa kelebihan seperti mengurangi biaya operasional layanan, kapabilitas tidak terbatas, masalah yang terpisah dan proses yang terisolasi [13].

### 3.6 Cloud Computing

*Cloud computing* merupakan sebuah model yang memungkinkan akses jaringan di mana-mana, nyaman, sesuai permintaan di sebuah *pool* komputasi yang terkonfigurasi [14]. *Cloud computing* memungkinkan akses server *on-demand* menjadi lebih mudah bagi banyak orang.

## 4. Result and Discussion

Bagian ini akan membahas mengenai hasil dari penelitian dan pengujian pada dua arsitektur yaitu LAMP Stack dan Jamstack, terdapat beberapa tipe pengujian pada penelitian yaitu performa, *load test*, dan kebutuhan biaya.

### 4.1 Migration

Migrasi adalah proses pertama yang perlu dilakukan sebelum melakukan pengujian dua arsitektur. Studi kasus yang digunakan adalah dengan melakukan migrasi *website landing page* Akunta.id dengan arsitektur LAMP Stack menjadi arsitektur Jamstack. Migrasi dilakukan dengan mengubah *framework* yang digunakan dari Laravel menjadi Next.js dan membangun *service* dari Laravel menjadi terpisah dan dihubungkan melalui API pada Next.js.

Migrasi server juga memiliki perbedaan spesifikasi, pada server LAMP Stack menggunakan server dengan spesifikasi yang dapat dilihat pada Tabel 1 sedangkan server CMS dan *hosting* Jamstack dapat dilihat pada Tabel 2 dan Tabel 3.

---

Tabel 1. Spesifikasi server LAMP Stack

Name	Value
Virtual CPU	2
Memory	2 GB
Storage	60 GB Disk

Tabel 2. Spesifikasi CMS API Jamstack

Name	Value
Virtual CPU	1

Memory 2 GB  
Storage 50 GB Disk

Tabel 3. Spesifikasi server Jamstack

Name	Value
Bandwidth	100 GB/Month
Serverless Functions	125,000 per site/Month
Build Minutes	300 Minutes/Month

#### 4. 2 Optimalisasi

Dua arsitektur *website* dilakukan optimalisasi terlebih dahulu sebelum dilakukan testing [15], optimalisasi yang dilakukan berupa *clear cache* dan *minified* pada *website* dengan arsitektur LAMP Stack dan *Image* yang ditaruh pada server berbeda khusus menangani media gambar pada CDN serta *asynchronous load* pada *website* yang menggunakan arsitektur Jamstack.

#### 4. 3 Performa

*Performance test* dilakukan menggunakan tiga *tools* yaitu Lighthouse dan GTmetrix digunakan untuk melakukan testing performa tiap *page* serta k6 yang digunakan untuk melakukan *load test*.

*Performance test* menggunakan detail *score* berdasarkan metode Lighthouse 8, terdapat enam penilaian yang memiliki jumlah bobot berbeda masing-masing penilaian. Hasil akhir dari testing akan menampilkan *score* dari nilai *percentage*.

Tabel 4. Detail nilai *score performance test*

Name of Metrics	Details	Score Percentage
First Contentful Paint	Seberapa cepat <i>text</i> dan gambar tampil pada saat <i>load</i> pertama kali sebuah <i>website</i> .	10%
Speed Index	Seberapa cepat <i>content</i> yang ada pada <i>website</i> termuat pada halaman <i>web</i> .	10%
Largest Contentfull	Seberapa cepat konten besar seperti <i>hero image</i> termuat pada <i>web</i> .	25%
Time to Interactive	Seberapa cepat halaman <i>web</i> benar-benar sudah <i>full</i> untuk dapat digunakan.	10%
Total Blocking Time	Seberapa lama halaman <i>web</i> di blok oleh <i>script</i> .	30%
Cumulative Layout Shift	Seberapa banyak bagian-bagian pada halaman <i>web</i> bergeser saat memuat halaman.	15%

Uji testing performa pada Lighthouse menghasilkan rasio yang tidak begitu signifikan namun tetap terlihat bahwa *website* yang menggunakan arsitektur Jamstack memiliki *score* yang lebih tinggi dari *website* yang menggunakan arsitektur LAMP Stack dilihat pada Tabel 5.

Tabel 5. Lighthouse *performance testing*

<b>Page of Website</b>	<b>Jamstack Score</b>	<b>LAMP stack Score</b>	<b>Ratio</b>
<i>Homepage</i>	74	61	1.21
<i>Fitur Dashboard</i>	80	63	1.26
<i>Fitur Digital Invoicing</i>	78	74	1.05
<i>Fitur Finance Report</i>	80	84	0.95
<i>Fitur Kelola Proyek</i>	79	67	1.17
<i>Fitur Penjualan</i>	82	62	1.32
<i>Fitur Project Budget</i>	78	66	1.18
<i>Fitur QR code Notification</i>	74	82	0.90
<i>Page Solusi UMKM</i>	81	74	1.09
<i>Page Bisnis Profesional</i>	76	67	1.13
<i>Page Penawaran</i>	74	82	0.90
<i>Page Support</i>	79	82	0.96
<i>Blog Post</i>	80	80	1.00
<b>TOTAL</b>	<b>1092</b>	<b>1013</b>	<b>1.07</b>

GTmetrix menghasilkan nilai rasio yang lebih signifikan dari Lighthouse dan tetap menghasilkan *score* bahwa *website* yang menggunakan arsitektur Jamstack lebih baik dari sisi performa dibanding dengan *website* menggunakan arsitektur LAMP Stack dilihat pada Tabel 6.

Tabel 6. GTmetrix *performance testing*

<b>Page of Website</b>	<b>Jamstack Score</b>	<b>LAMP stack Score</b>	<b>Ratio</b>
<i>Homepage</i>	96	74	1.29
<i>Fitur Dashboard</i>	100	71	1.40
<i>Fitur Digital Invoicing</i>	100	71	1.40
<i>Fitur Finance Report</i>	100	59	1.69
<i>Fitur Kelola Proyek</i>	100	78	1.28
<i>Fitur Penjualan</i>	100	75	1.33
<i>Fitur Project Budget</i>	100	65	1.53
<i>Fitur QR code Notification</i>	100	73	1.36
<i>Page Solusi UMKM</i>	100	65	1.53
<i>Page Bisnis Profesional</i>	100	80	1.25
<i>Page Penawaran</i>	100	82	1.21
<i>Page Support</i>	100	82	1.21
<i>Blog Post</i>	100	84	1.19
<b>TOTAL</b>	<b>1396</b>	<b>1045</b>	<b>1.33</b>

*Load testing* pada k6 menggunakan beberapa *stage* yang ditujukan untuk melakukan simulasi *user* mengakses *website* pada server secara bersamaan, tiap *stage* menghasilkan virtual user berbeda-beda dan signifikan.

Tabel 7. Durasi dan jumlah *user* tiap stage k6

<b>Durasi</b>	<b>Virtual User</b>
10 detik	20 <i>virtual user</i>
20 detik	800 <i>virtual user</i>
5 detik	0 <i>virtual user</i>
5 detik	200 <i>virtual user</i>
1 detik	50 <i>virtual user</i>
5 detik	0 <i>virtual user</i>
15 detik	500 <i>virtual user</i>

*Load testing* k6 menghasilkan nilai rasio yang sangat signifikan. *Website* dengan arsitektur Jamstack memiliki *request failed* yang jauh lebih sedikit dibandingkan dengan *website* dengan arsitektur LAMP Stack, hal ini dapat terjadi dikarenakan *server* Jamstack adalah serverless sedangkan *server* LAMP Stack adalah berbasis Linux VPS biasa, terlihat pada Tabel 8.

Tabel 8. k6 load testing

<b>Page of Website</b>	<b>Jamstack Failed Request</b>	<b>LAMP stack Failed Request</b>	<b>Ratio</b>
<i>Homepage</i>	2	147	73.5
<i>Fitur Dashboard</i>	11	435	39.54
<i>Fitur Digital Invoicing</i>	12	463	38.58
<i>Fitur Finance Report</i>	11	601	54.63
<i>Fitur Kelola Proyek</i>	9	522	58
<i>Fitur Penjualan</i>	14	647	46.21
<i>Fitur Project Budget</i>	13	590	45.38
<i>Fitur QR code Notification</i>	2	567	283.5
<i>Page Solusi UMKM</i>	4	605	151.25
<i>Page Bisnis Profesional</i>	9	647	71.88
<i>Page Penawaran</i>	11	492	44.72
<i>Page Penawaran</i>	2	660	330
<i>Page Support</i>	6	964	482
<i>Blog Post</i>	188	759	4.03
<b>TOTAL</b>	<b>294</b>	<b>8099</b>	<b>27.54</b>

#### 4.4 Cost Effectiveness

Biaya dua arsitektur yang berbeda pada penelitian memiliki total kebutuhan biaya yang berbeda juga, masing-masing arsitektur tetap membutuhkan biaya *server*. Arsitektur Jamstack pada penelitian membutuhkan API pihak ketiga dengan biaya yang gratis dengan limitasi yang wajar sedangkan arsitektur LAMP Stack tidak memiliki biaya pada API karena sudah menjadi satu server, dapat dilihat pada Tabel 9.

Tabel 9. Cost effectiveness

<b>Name</b>	<b>LAMP stack</b>	<b>Jamstack</b>
<i>Server Hosting</i>	Rp500,000	Rp0
<i>CMS Hosting</i>	Rp0	Rp179,938.20
<i>Search API (Algolia)</i>	-	Rp0
<b>TOTAL</b>	<b>Rp500,000</b>	<b>Rp179,938.20</b>

#### 5. Conclusion

Penelitian ini menggunakan dua arsitektur bernama LAMP Stack dan Jamstack yang memiliki perbedaan dari metode pengembangan dan *tools* yang digunakan, LAMP Stack biasanya menggunakan *tools open source* dan Jamstack menggunakan *tools* pihak ketiga ataupun membangun API sendiri. Migrasi dari arsitektur LAMP Stack menuju Jamstack membutuhkan pembuatan kode ulang dan API berbasis dari sumber kode, dikarenakan dua arsitektur sangat berbeda dari sisi *environment* hingga kode. Optimalisasi dilakukan pada masing-masing arsitektur berguna untuk memaksimalkan hasil *score* pada *test* yang dijalankan. Hasil testing menghasilkan *score* bahwa arsitektur Jamstack unggul dari sisi performa namun tidak begitu signifikan dikarenakan LAMP Stack juga memiliki fitur *cache* yang dapat mempercepat akses tiap halaman, namun dari sisi *load test*, Jamstack memiliki *score* yang sangat signifikan lebih baik dibanding LAMP Stack dikarenakan menggunakan serverless pada *hosting website*, dan biaya yang dibutuhkan untuk membangun *website* menggunakan arsitektur Jamstack relatif lebih murah dibanding *website* LAMP Stack dikarenakan rata-rata *vendor* memberikan akses gratis yang lebih banyak pada arsitektur JAMStack.

Saran dari penelitian ini adalah untuk dapat menggunakan CMS pihak ketiga sehingga tidak memerlukan biaya *hosting server* sehingga membangun *website* menggunakan

arsitektur Jamstack dapat dilakukan secara gratis sepenuhnya. Penelitian lebih lanjut juga diperlukan dikarenakan popularitas Jamstack masih kurang dibanding LAMP Stack walaupun Jamstack memiliki fitur yang relatif lebih baik dibanding LAMP Stack.

### References

- [1] W. Hardinugraha, O. Sudana, and D. P. Githa, "Web-Based E-Marketing Application for Balinese Art Studio," *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, pp. 34–45, 2021.
  - [2] R. Y. He, "Design and Implementation of Web Based on Laravel Framework," *Proceedings of the 2014 International Conference on Computer Science and Electronic Technology*, vol. 6, no. Iccset 2014, pp. 301–304, 2015, doi: 10.2991/iccset-14.2015.66.
  - [3] T. Hoang, "JAMStack Continuous Integration and Continuous Deployment with CircleCI and Netlify," 2020.
  - [4] A. Karanjit, "MEAN vs. LAMP Stack," 2016, Accessed: Aug. 10, 2022. [Online]. Available: [https://repository.stcloudstate.edu/csit\\_etds/11](https://repository.stcloudstate.edu/csit_etds/11)
  - [5] U. v. Ramana and T. v. Prabhakar, "Some experiments with the performance of LAMP architecture," *Proceedings - Fifth International Conference on Computer and Information Technology, CIT 2005*, vol. 2005, no. June, pp. 916–920, 2005, doi: 10.1109/CIT.2005.169.
  - [6] M. Biilmann, *Modern Web Development on the JAMstack*. O'Reilly Media, Incorporated, 2019.
  - [7] P. K. Dey and J. Kinch, "Risk management in information technology projects," *International Journal of Risk Assessment and Management*, vol. 9, no. 3, pp. 311–329, 2008, doi: 10.1504/IJRAM.2008.019747.
  - [8] K. S. Wibawa and I. N. Piarsa, "Secure Data Monitoring System with Encrypt Data Transmission over Radio Communication Based on Microcontroller," *International Journal of Computer Applications*, vol. 179, no. 21, pp. 19–22, 2018.
  - [9] R. Dhuny, A. A. Iqbal Peer, N. A. Mohamudally, and N. Nissanke, "Performance evaluation of a portable single-board computer as a 3-tiered LAMP stack under 32-bit and 64-bit Operating Systems," *Array*, p. 100196, Sep. 2022, doi: 10.1016/j.array.2022.100196.
  - [10] R. C. James, *Mathematics dictionary*. Springer Science & Business Media, 1992.
  - [11] N. J. Yeager and R. E. McGrath, *Web server technology*. Morgan Kaufmann, 1996.
  - [12] G. Adzic and R. Chatley, "Serverless computing: economic and architectural impact," pp. 884–889, 2017, doi: 10.1145/3106237.3117767.
  - [13] C. Boulton, "Serverless: The future of cloud computing?," *CIO.com*, pp. 1–3, 2018, [Online]. Available: <https://www.cio.com.au/article/631702/serverless-future-cloud-computing/>
  - [14] Z. Sun, "The Calculus of Cloud Computing The Calculus of Cloud Computing," no. February, 2019, doi: 10.13140/RG.2.2.13483.49446/1.
  - [15] R. Muliono, "Website Optimization Cms Based on Google Pegaspeed," *Computer Engineering, Science and System Journal*, vol. 1, no. 2, pp. 32–35, 2016, doi: 10.24114/cess.v1i2.4065.
-