

## Simple Additive Weighting untuk Front-end Framework Terbaik

Lestari Yusuf<sup>a,1</sup>, Taufik Hidayatulloh<sup>b,2,\*</sup>, Dini Nurlaela<sup>b,3</sup>, Lila Dini Utami<sup>b,4</sup> dan Fuad Nur Hasan<sup>b,5</sup>

<sup>a</sup> Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri, Jln. Jatiwaringin Raya No. 02 RT.08 RW.013 Kelurahan Cipinang Melayu Kecamatan Makassar, Jakarta Timur, 13620, Indonesia

<sup>b</sup> Universitas Bina Sarana Informatika, Jl. Kramat Raya No. 98, Senen, Jakarta Pusat 10450, Indonesia

<sup>1</sup> lestari.lyf@nusamandiri.ac.id; <sup>2</sup> taufik.tho@bsi.ac.id; <sup>3</sup> dini.dur@bsi.ac.id; <sup>4</sup> lila.ldu@bsi.ac.id; <sup>5</sup> fuad.fnu@bsi.ac.id  
\*corresponding author

### INFORMASI ARTIKEL

Diterima : 26 Juni 2020  
Diulas : 23 Juli 2020  
Direvisi : 29 Juli 2020  
Diterbitkan : 27 Agustus 2020

#### KataKunci:

Framework Terbaik  
Front-end Framework  
Simple Additive Weighting

### ABSTRAK

Aplikasi *web* berbasis dekstop beberapa tahun terakhir ini mendapatkan permintaan besar yang menginginkan area ini agar lebih canggih dan kompleks, bukan hanya itu tetapi pengguna menginginkan aplikasi *web* dapat dijalankan di *mobile device*. Tampilan *web* yang hanya didesain untuk perangkat komputer menyebabkan pengguna kesulitan saat membuka tampilan sebuah halaman *web* di device yang berbeda. Melalui *framework front-end*, *web developer* membantu pengguna membuat program menjadi lebih responsif dan dapat dijalankan diberbagai open source seperti *windows*, *ios*, dan *android*. Sistem pengambilan keputusan yang dapat menentukan *framework front-end* terbaik bisa menjadi alternatif pemecahan masalah para *web developer* dalam menentukan *framework front-end* mana yang lebih mudah dan nyaman untuk digunakan. *Simple Additive Weighting* digunakan untuk menganalisa dan memutuskan alternatif mana yang terbaik dengan perhitungan yang mengambil lima kriteria utama dalam penelitian ini yaitu *Preprocessor*, *Responsive*, *Browser Support*, *Easy to Use* dan *Template*. Dalam penelitian ini prefensi tertinggi diperoleh oleh Bootstrap 1,000 sedangkan untuk Foundation dan Bulma mendapatkan prefensi sebesar 0,868 dan 0,820.

#### Keywords:

Best Framework  
Front-end Framework  
Simple Additive Weighting

### ABSTRACT

Web applications that used to run on a desktop in recent years have received huge demand for this area to be more sophisticated and complex, not only that, but users also want web applications to run on mobile devices. Web appearance that is only designed for computer devices will make users difficult when opening a web page display on a different device. By using the CSS framework library, web developers will be greatly helped in making the program more responsive and can also be run on a variety of Open Source both Windows, iOS, and Android. Decision-making system that can determine the best front-end Framework can be an alternative solution for web developers to determine which front-end framework is easier and more convenient to use. Simple Additive Weighting is used to analyze and decide which the best alternative with calculations that take five main criteria in this research that is Preprocessor, Responsive, Browser Support, Easy to Use, and Template. In this study the highest preference was obtained by Bootstrap 1,000 while Foundation and Bulma received preference of 0.868 and 0.820.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



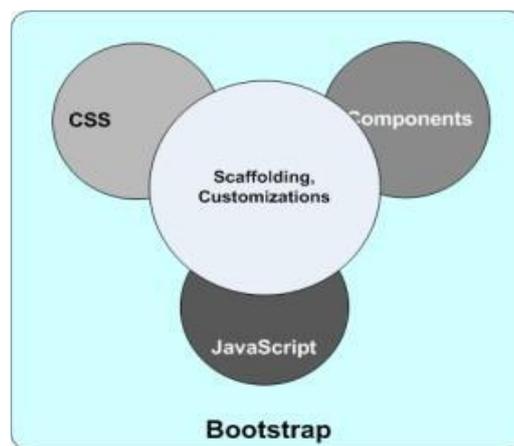
## I. Pendahuluan

Aplikasi *web* yang biasa dijalankan di sebuah dekstop beberapa tahun terakhir ini mendapatkan permintaan besar yang menginginkan area ini agar lebih canggih dan kompleks, bukan hanya itu tetapi pengguna menginginkan aplikasi *web* dapat dijalankan di *mobile device* [1]. Banyaknya keluaran perangkat *mobile* yang baru dengan ukuran layar yang berbeda-beda menjadikan masalah untuk para *developer web*, yang akhirnya membutuhkan sebuah desain *web* yang responsif. Tujuan dari desain yang responsif adalah untuk membangun halaman *web* yang dapat mendeteksi ukuran layar pengunjung sebuah *web* dan diikuti dengan perubahan layoutnya [2].

Salah satu diantara faktor utama yang harus direncanakan dengan baik agar sebuah *website* dapat berjalan dengan tampilan yang tetap rapih, cepat dan dapat diakses oleh berbagai macam *browser* dan perangkat lunak yang berbeda-beda adalah melakukan rancang bangun dari tampilan sebuah *web* [3]. Beberapa model *Framework* mampu menjawab keinginan para pengguna *website* dan developer yang memiliki masalah pengembangan sebuah *website* dan desain. Terdapat banyak sekali *framework CSS* yang dapat membantu *developer web* [4]. Pada penelitian ini membandingkan *Framework Bulma* yang baru saja release pada tahun 2018 yang mendapatkan *react* 39.779 *star* (<https://github.com/jgthms/bulma>). *Bulma* memiliki *Flexbox* yang merupakan unggulan didalam *framework*, *framework* yang didirikan 2018 ini menyajikan *framework front-end* yang menyediakan berbagai fitur dan komponen yang dibutuhkan [5].

Selanjutnya *Framework Foundation* yang mendapatkan *react* pada *github.com* sebanyak 28.265 *star* (<https://github.com/foundation/foundation-sites>). Kerangka kerja yang didirikan oleh ZURB di California ini menggunakan *SASS* sebagai *pre-processor* yang membuat tulisan terorganisasi dengan baik dan kerangka kerja ini mempermudah para developer web dengan mengutamakan responsifnya [6].

Sejak 2011 *Framework Bootstrap* yang dirilis mendapatkan reaksi pada *github.com* sekitar 140.848 *star*. *Bootstrap* merupakan *framework* yang digunakan untuk membangun tatap muka para *web developer*, dibangun dengan menggabungkan *java script* dan *css bootstrap* memenuhi kebutuhan komponen yang penting untuk para developer [7]. Terhitung 48 *framework* yang terdapat di *github.com* sejak pertama kali *framework CSS* ditemukan diantara tahun 2000 sampai 2018. Pada tahun 2019 terdapat 4 *framework* terbaru. Berikut struktur *bootstrap* yang diperlihatkan pada Gambar 1.



Gambar 1. Struktur *Bootstrap*

Semakin meningkatnya *framework library CSS* yang bermunculan maka, pemilihan *framework library CSS* dapat membantu para *developer web* atau para programmer muda untuk memilih *framework library CSS* terbaik untuk digunakan. Penelitian sebelumnya telah dilakukan pengulasan perbedaan antara *framework front-end CSS* [2]. Beberapa penelitian telah menampilkan beberapa *front-end framework* yang dapat membuat proses pembuatan sebuah *web* lebih mudah yang berfokus pada analisa yang efektif antara *Bootstrap* dan *Foundation* [8].

Pengujian sensitifitas bertujuan membantu pengambilan keputusan berdasarkan nilai alternatif terbaik menggunakan metode *Multiple Attribute Decision Making* (MDAM) yang mana membandingkan antara *weighted product* (WP). *Simple Additive Weighting* dapat digunakan sebagai penentuan nilai tertinggi untuk kelayakan angkutan umum [8]. Pada penelitian sebelumnya metode SAW digunakan untuk melakukan seleksi anggota tari [9], selain itu pada penelitian sebelumnya *Simple additive weighting* (SAW) digunakan untuk memutuskan penerima kartu indonesia pintar [10]. Penelitian ini melakukan perhitungan dari setiap bobot kriteria yang diambil agar mendapatkan hasil berupa perankingan dari setiap *framework front end Bootstrap*, *Foundation* dan *Bulma* dalam mendapatkan *framework front end* yang terbaik.

## II. Metode

Pada beberapa penelitian sebelumnya, metode SAW sering dikenal sebagai metode penjumlahan dari sebuah bobot. *Simple Additive Weighting* (SAW) merupakan metode yang dianggap sangat intuisi dan memiliki cara yang mudah dalam menyelesaikan masalah pengambilan keputusan dengan banyak kriteria [11]. Proses normalisasi matriks keputusan pada metode SAW merupakan tahap untuk mendapatkan nilai alternatif ( $A_i$ ) yang terpilih dengan cara menghitung skala dan membandingkannya dengan nilai tertinggi dari seluruh alternatif yang ada [12]. Metode penelitian ini dilakukan untuk menentukan *framework Front-end* terbaik seperti yang dijelaskan pada Gambar 2.



Gambar 2. Metode Penelitian Pemilihan *Framework Front-end* Terbaik

Langkah pertama yang harus dilakukan untuk menyelesaikan metode SAW ini adalah menentukan kriteria yang akan digunakan dan ditentukan jenis kriteria termasuk kedalam *benefit* atau *cost* [13]. Kriteria yang digunakan dijabarkan pada Tabel 1.

Tabel 1. Kriteria

C1	<i>Preprocessor</i>	<i>benefit</i>
C2	<i>Responsive</i>	<i>benefit</i>
C3	<i>Browser Support</i>	<i>benefit</i>
C4	<i>Easy to use</i>	<i>benefit</i>
C5	<i>Tamplate/layout</i>	<i>benefit</i>

Langkah berikutnya adalah pencarian bobot untuk setiap kriteria. Pencarian bobot kriteria ini akan membandingkan matriks dari setiap kriteria dengan objektif menggunakan perhitungan *fuzzy* skala 1-5 yang hasil perhitungan bobotnya diperoleh dari para pengambil keputusan dan dijelaskan pada Tabel 2.

Tabel 2. Nilai Bobot Setiap Kriteria

<i>Preprocessor</i>	0.190
<i>Responsive</i>	0.290
<i>Browser support</i>	0.340
<i>Easy to Use</i>	0.110
<i>Tamplate/layout</i>	0.070

Setelah mendapatkan bobot selanjutnya adalah proses pembuatan matriks keputusan berdasarkan kriteria kemudian dinormalisasi [14]. Berikut perhitungan proses perankingan:

a. Formula untuk normalisasi menggunakan persamaan (1)

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max } x_{ij}}, & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min } x_{ij}}{x_{ij}}, & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases} \quad (1)$$

Dimana :

$r_{ij}$  = Rating kinerja normalisasi

$\text{Max } x_{ij}$  = nilai maksimum dari setiap baris dan kolom

$\text{Min } x_{ij}$  = nilai minimum dari setiap baris dan kolom

$x_{ij}$  = baris dan kolom matriks

$i = 1, 2, 3, \dots, m$

$j = 1, 2, 3, \dots, n$

b. Formula untuk menghitung nilai preferensi

Nilai preferensi untuk setiap alternatif menggunakan persamaan (2)

$$v_i = \sum_{j=1}^n w_j r_{ij} \quad (2)$$

Dimana :

$v_i$  = nilai akhir dari alternatif

$w_i$  = bobot yang telah ditentukan

### III. Hasil dan Pembahasan

Perhitungan di atas menjelaskan kepentingan setiap kriteria dan yang mendapatkan bobot yang lebih besar menandakan bahwa kriteria tersebut lebih penting. Setelah mendapat bobot setiap kriteria, langkah berikutnya yang harus dilakukan adalah memberikan nilai untuk setiap kriteria.

Rata-rata untuk nilai Preprocessor dibagi kedalam 3 angka *fuzzy*. Nilai yang akan dijadikan bobot merupakan bilangan *fuzzy* yang di ubah kedalam bilangan *crisp*, dimana bilangan *crisp* sendiri memiliki makna teori dimana jika angka mendekati 0 maka memiliki tingkat ketergantungan yang rendah, dan jika angka mendekati 1 maka tingkat ketergantungan menjadi lebih tinggi [15] dan untuk kriteria Preprocessor memiliki 3 angka *fuzzy* yaitu Kurang Baik (KB), Baik (B) dan Sangat Baik (SB) yang diperlihatkan pada Tabel 3.

Tabel 3. Nilai Kriteria Preprocessor

Rentang nilai	Angka Fuzzy	Nilai
Tidak Ada Preprocessor	Kurang Baik	$0/(3-1)=0$
Hanya Less or sass	Baik	$1/(3-1)=0,5$
Less&Sass	Sangat Baik	$2/(3-1)=1$

Rata-rata untuk nilai kriteria *Responsive* dibagi kedalam 2 angka *Fuzzy* yaitu Tidak dan Ya. Nilai yang dihasilkan diperlihatkan pada Tabel 4.

Tabel 4. Nilai Kriteria Responsive

Rentang Nilai	Angka Fuzzy	Nilai
tidak responsive	no	0.000
responsive	yes	1.000

Kriteria yang memiliki nilai bobot paling tinggi, rata-rata *browser support* dibagi kedalam 5 angka *fuzzy*, Sangat Buruk (SB), Buruk(B), Cukup Baik(CB), Baik(B) dan Baik Sekali(BS). Nilai untuk *Browser Support* dijabarkan pada Tabel 5 dibawah ini.

Tabel 5. Nilai Kriteria Browser Support

Rentang Nilai	Angka Fuzzy	Nilai
0 Browser dan Os	very bad	$0/4=0$

Rentang Nilai	Angka Fuzzy	Nilai
1-3 Browser dan OS	bad	1/4=0.250
4-5 Browser dan OS	good enough	2/4=0.500
4-6 Browser dan OS	good	3/4=0.75
6 - n Browser dan OS	very good	4/4=1

Kriteria *Easy To Use* di berikan bobot yang tidak begitu tinggi karena pada dasarnya hampir semua *Front-end Framework* mudah digunakan. Pada penelitian ini rata-rata nilai untuk kriteria *Easy to Use* dibagi ke dalam 5 angka *Fuzzy*. Angka tersebut dijelaskan pada Tabel 6.

Tabel 6. Nilai Kriteria *Easy to Use*

Rentang Nilai	Angka Fuzzy	Nilai
0-30	very hard	0/4=0
31-40	hard	1/4=0.250
41-68	easy enough	2/4=0.500
69-80	easy	3/4=0.75
81-100	very easy	4/4=1

Kriteria *Easy to Use*, kriteria *Template* atau *Layout* pun dibagi kedalam 5 angka *fuzzy*, yaitu Buruk Sekali (BS), Buruk (B), Cukup Baik (CB), Baik (B) dan Baik Sekali (BS). Data dideskripsikan pada Tabel 7.berikut:

Tabel 7.Nilai Kriteria Template/Layout

Rentang Nilai	Angka Fuzzy	Nilai
0-30	very bad	0/4=0
31-40	bad	1/4=0.250
41-68	good enough	2/4=0.500
69-80	good	3/4=0.75
81-100	very good	4/4=1

Setelah itu, langkah berikutnya adalah mendapatkan data nilai yang memenuhi dari setiap alternatif yang akan dibandingkan. Data diisi oleh para pengambil keputusan yang dijelaskan pada Tabel 8.

Tabel 8. Nilai Alternatif untuk setiap Kriteria

Pembobotan Alternatif	C1	C2	C3	C4	C5
A1 (Foundation)	Sass	Ya	1.000	68	75
A2 (Bootstrap)	Less dan Sass	Ya	1.000	80	80
A3 (Bulma)	Less	Ya	0.750	78	78

Angka-angka tersebut dikonversi kedalam bilangan kedalam bilangan matriks data akan dideskripsikan pada matriks.

$$\begin{bmatrix} 0.500 & 1.000 & 1.000 & 0.500 & 0.750 \\ 1.000 & 1.000 & 1.000 & 0.750 & 0.750 \\ 0.500 & 1.000 & 0.750 & 0.750 & 0.750 \end{bmatrix}$$

Setelah melakukan konversi bilangan, maka langkah berikutnya adalah melakukan perhitungan normalisasi untuk setiap alternatif. Pada tahap ini, jika bobot dari kriteria semakin besar semakin baik, maka gunakan perhitungan kriteria benefit [11].

#### Alternatif 1

$$r_{11} = \frac{0.500}{\max\{0.500; 1.000; 0.500\}} = \frac{0.500}{1.000} = 0.500$$

$$r_{12} = \frac{1.000}{\max\{1.000; 1.000; 1.000\}} = \frac{1.000}{1.000} = 1.000$$

$$r_{13} = \frac{1.000}{\max\{1.000; 1.000; 0.750\}} = \frac{1.000}{1.000} = 1.000$$

$$r_{14} = \frac{0.500}{\max\{0.500; 0.750; 0.750\}} = \frac{0.500}{0.750} = 0.667$$

$$r_{15} = \frac{0.750}{\max\{0.750; 0.750; 0.750\}} = \frac{0.750}{0.750} = 1.000$$

### Alternatif 2

$$r_{21} = \frac{1.000}{\max\{0.500; 1.000; 0.500\}} = \frac{1.000}{1.000} = 1.000$$

$$r_{22} = \frac{1.000}{\max\{1.000; 1.000; 1.000\}} = \frac{1.000}{1.000} = 1.000$$

$$r_{23} = \frac{1.000}{\max\{1.000; 1.000; 0.750\}} = \frac{1.000}{1.000} = 1.000$$

$$r_{24} = \frac{0.750}{\max\{0.500; 0.750; 0.750\}} = \frac{0.750}{0.750} = 1.000$$

$$r_{25} = \frac{0.750}{\max\{0.750; 0.750; 0.750\}} = \frac{0.750}{0.750} = 1.000$$

### Alternatif 3

$$r_{31} = \frac{0.500}{\max\{0.500; 1.000; 0.500\}} = \frac{0.500}{1.000} = 0.500$$

$$r_{32} = \frac{1.000}{\max\{1.000; 1.000; 1.000\}} = \frac{1.000}{1.000} = 1.000$$

$$r_{33} = \frac{0.750}{\max\{1.000; 1.000; 0.750\}} = \frac{0.750}{1.000} = 0.750$$

$$r_{34} = \frac{0.750}{\max\{0.500; 0.750; 0.750\}} = \frac{0.750}{0.750} = 1.000$$

### Normalisasi Matriks

$$\begin{bmatrix} 0.500 & 1.000 & 1.000 & 0.667 & 1.000 \\ 1.000 & 1.000 & 1.000 & 1.000 & 1.000 \\ 0.500 & 1.000 & 0.750 & 1.000 & 1.000 \end{bmatrix}$$

Setelah dilakukan perhitungan antara bobot setiap kriteria ( $w$ ) dan normalisasi matrik hasil perhitungandari setiap alternatif ( $r$ ) untuk mendapatkan nilai prevensi ( $vi$ ). berikut Nilai  $w$  yang digunakan untuk setiap bobot kriteria:

$$w = [0.190 \ 0.290 \ 0.340 \ 0.110 \ 0.070]$$

$$V1 = (0.500 \times 0.190) + (1.000 \times 0.290) + (1.000 \times 0.340) + (0.667 \times 0.110) + (1.000 \times 0.070) = 0.868$$

$$V2 = (1.000 \times 0.190) + (1.000 \times 0.290) + (1.000 \times 0.340) + (1.000 \times 0.110) + (1.000 \times 0.070) = 1.000$$

$$V3 = (0.500 \times 0.190) + (1.000 \times 0.290) + (0.750 \times 0.340) + (1.000 \times 0.110) + (1.000 \times 0.070) = 0.820$$

Nilai tertinggi dari tiga alternatif tersebut dari penjumlahan matriks. Peringkat ditunjukkan pada Tabel 9.

Tabel 9. Peringkat *Front-end Framework*

Alternatif	Nilai prefensi ( $V_i$ )	Ranking
A1(Foundation)	0.862	2
A2(Bootstrap)	1.000	1
A3(Bulma)	0.820	3

Setiap bobot kriteria dari alternatif dan dihitung menggunakan metode *Simple Additive Weighting* (saw) maka hasil dari penelitian ini akan memudahkan *web developer* dalam hal pemilihan *front-end framework* terbaik yang akan digunakan atau bahkan yang berencana akan mengkombinasikan *front-end framework* tersebut, karena kriteria-kriteria pada penelitian ini sudah dapat mewakili kebutuhan-kebutuhan *front-end framework* yang disesuaikan dengan *web developer*. Sementara kekurangan yang ada pada penelitian ini terdapat pada alternatif yang digunakan hanya untuk tiga alternatif *front-end framework* saja.

#### IV. Kesimpulan dan Saran

Perhitungan yang dilakukan menggunakan metode *Simple Additive Weighting* dengan menggunakan lima kriteria yang jika bobot kriterianya diurutkan akan didapat bobot terbesarnya adalah kriteria *browser support, responsive, preprocessor, easy to use* dan yang terakhir adalah *template/layout*, maka diperoleh urutan *front-end framework* terbaik, yang pertama yaitu *bootstrap* yang mendapatkan nilai prefensi sebesar 1.000, *Foundation* dengan nilai prefensi sebesar 0.868 dan *Bulma* mendapatkan nilai prefensi sebesar 0.820. Pada penelitian berikutnya kami menyarankan agar lebih memperdalam kriteria-kriteria dari *framework front-end* tersebut. Penelitian dapat dilakukan menggunakan metode-metode pengambilan keputusan lain, serta mengkomparasi dengan metode yang lainnya.

#### Daftar Pustaka

- [1] B. Vukelic, "Comparison of front-end frameworks for web applications development 4," *Zb. Veleuc. u rijeci*, vol. 6, no. 1, pp. 261–282, 2018.
- [2] N. Jain, "Review Of Different Responsive CSS Front-End Frameworks," *J. Glob. Res. Comput. Sci.*, vol. 5, no. 11, pp. 5–10, 2014.
- [3] A. Zakir, "Rancang bangun responsive web layout dengan menggunakan bootstrap framework," *J. Nas. Inform. dan Teknol. Jar.*, vol. 1, no. 1, pp. 7–10, 2016.
- [4] J. O. Meiert, *The Little Book of HTML / CSS Frameworks*. United States of America: O'Reilly Media, 2015.
- [5] G. Gunadi, "Pengembangan aplikasi web berbasis flexbox untuk pengelolaan jadwal produksi dan dokumen digital menggunakan framework bulma di PT. Gramedia," *Infotech*, vol. 5, no. 2, pp. 40–47, 2019.
- [6] S. Anugerah, "Pemodelan responsive web menggunakan foundation framework dalam pengembangan perangkat lunak," vol. 2013, no. semnasIF, pp. 230–236, 2013.
- [7] B. Jacobus and J. Marah, *Mastering Bootstrap 4*, Second. Brimingham: PACKT Publishing Ltd., 2018.
- [8] N. S. Nekida, R. A. Saputra, L. S. Ramdhani, and T. Hidayatulloh, "implementasi e-ticketing uji kelayakan kendaraan bermotor untuk meningkatkan pelayanan publik," *J. SWABUMI*, vol. 6, no. 2, pp. 143–148, 2018.
- [9] H. Adela, K. A. Jasmi, B. Basiron, M. Huda, and A. Maseleno, "Selection of dancer member using simple additive weighting," *Int. J. Eng. Innov. Technol.*, no. June, 2018.
- [10] A. Saryoko, S. Muttaqin, and R. Hidayat, "sistem penunjang keputusan penerima kartu indonesia pintar menggunakan metode simple additive weighting ( SAW )," *J. Tek. Komput.*, vol. V, no. 2, pp. 139–146, 2019.
- [11] G.-H. Tzeng and J.-J. H. Huang, *Multiple Attribute Decision Making Methods And Applications*. New York: Taylor & Francis Group, an Informa business., 2011.
- [12] M. A. Mude, "PADA KASUS UMKM," vol. 8, no. Agustus, pp. 76–81, 2016.
- [13] D. F. Shiddieq and E. Septyan, "analisis perbandingan metode ahp dan saw dalam penilaian kinerja karyawan (studi kasus di PT. Grafindomedia Pratama Bandung)," *J. LPKIA*, vol. 10, no. 2, 2017.
- [14] N. C. Resti, "Penerapan metode simple additive weighting ( saw ) pada sistem pendukung keputusan pemilihan lokasi untuk Cabang Baru Toko," *J. INTENSIF*, vol. 1, no. 2, pp. 102–107, 2017.
- [15] S. Suhada, T. Hidayatulloh, and S. Fatimah, "Penerapan fuzzy madm model weighted product dalam pengambilan keputusan kelayakan penerimaan kredit di BPR Nusamba Sukaraja ( The Application of Fuzzy MADM Model Weighted Product in Decisions Support of Credit Worthiness in the BPR Nusamba Sukaraja )," *JUITA*, vol. VI, pp. 61–71, 2018.