

---

# PENERAPAN METODE TOPSIS DALAM PEMILIHAN *GUEST STAR* KONSER MUSIK DI YOGYAKARTA

Marta Y (215314154), Victoria A F S (215314158),  
Resiana K J (215314159), Loadtriani O S (215314172)

**Abstrak** - Keberhasilan suatu konser musik dapat dipengaruhi oleh kehadiran seorang bintang tamu (*guest star*) yang dipilih dengan cermat. Kehadiran *guest star* tidak hanya diharapkan dapat meningkatkan profitabilitas bagi pihak penyelenggara, melainkan juga memberikan pengalaman istimewa dan kepuasan kepada para penonton. Oleh karena itu, dalam rangka membantu proses pemilihan *guest star* untuk konser musik di Yogyakarta, penelitian ini mengembangkan sebuah Sistem Pendukung Keputusan (SPK). SPK ini dirancang untuk memfasilitasi pengambilan keputusan dengan memanfaatkan data yang telah diolah secara relevan. Metode SPK yang diterapkan dalam penelitian ini adalah metode Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). Dengan menggunakan metode ini, penyelenggara dapat menetapkan nilai bobot untuk setiap kriteria yang relevan, yang kemudian digunakan dalam proses perankingan untuk menentukan alternatif terbaik dari sejumlah pilihan *guest star*. Hasil perhitungan metode TOPSIS ini menghasilkan nilai preferensi untuk setiap alternatif, yang nantinya digunakan sebagai landasan dalam pengambilan keputusan terkait pemilihan *guest star*.

**Kata kunci** : *Guest Star, Sistem Pendukung Keputusan, TOPSIS*

**Abstract** - The success of a music concert can be influenced by the presence of a carefully selected guest star. The presence of guest stars is not only expected to increase profitability for the organizers but also provide a special experience and satisfaction to the audience. Therefore, to assist the process of selecting guest stars for music concerts in Yogyakarta, this research developed a Decision Support System. This DSS is designed to facilitate decision-making by utilizing data that has been processed in a relevant manner. The SPK method applied in this research is the Technique for Order of Preference by Similarity to the Ideal Solution (TOPSIS) method. By using this method, the organizer can assign a weight value to each relevant criterion, which is then used in the role process to determine the best alternative from many guest star choices. The results of the TOPSIS method calculation produce a preference value for each alternative, which will later be used as a basis for making decisions regarding the selection of guest stars.

**Keyword** : *Guest Star, Decision Support System, TOPSIS*

## I. PENDAHULUAN

Dalam dunia hiburan yang terus berkembang, konser musik menjadi acara yang digemari oleh masyarakat. Kesuksesan sebuah konser tidak hanya bergantung pada penyelenggaraan acara yang baik, tetapi juga pada kehadiran *guest star* yang dapat memberikan kesan istimewa bagi penonton. Oleh karena itu, pemilihan *guest star* yang tepat menjadi faktor krusial dalam menentukan keberhasilan sebuah konser.

Untuk membantu proses pengambilan keputusan dalam pemilihan *guest star*, Sistem Pendukung Pengambilan Keputusan (SPK) hadir sebagai solusi. Keputusan ini harus mempertimbangkan anggaran yang tersedia agar acara tetap menguntungkan. Selain itu, pemilihan *guest star* juga harus memikirkan profil audiens yang dituju, sehingga dapat menarik perhatian dan minat mereka. Penelitian ini bertujuan merancang dan mengimplementasikan SPK Pemilihan *Guest Star* untuk Konser di Yogyakarta. Metode TOPSIS (*Technique for Order of*

*Preference by Similarity to Ideal Solution*) dipilih sebagai dasar metode untuk mengelola data dan memberikan rekomendasi *guest star* yang optimal.

Keberhasilan sebuah konser tidak hanya dilihat dari jumlah penonton, tetapi juga dari kepuasan dan respons positif yang diberikan oleh para penonton. Oleh karena itu, peran SPK dengan metode TOPSIS diharapkan dapat membantu penyelenggara konser dalam memilih *guest star* yang tidak hanya sesuai dengan tema acara tetapi juga dapat memberikan pengalaman hiburan yang tak terlupakan. Tujuan utama adalah agar acara tidak hanya mencapai profitabilitas, tetapi juga menciptakan pengalaman yang memuaskan bagi audiens sehingga mereka akan kembali untuk acara berikutnya.

Melalui penelitian ini, diharapkan dapat memberikan kontribusi dalam pengembangan sistem pendukung pengambilan keputusan yang dapat digunakan oleh industri hiburan, khususnya dalam hal pemilihan *guest star* untuk konser di Jogja. Penerapan metode TOPSIS diharapkan mampu memberikan rekomendasi *guest star* yang optimal dan mendukung kesuksesan konser secara keseluruhan.

## II. METODOLOGI PENELITIAN

### A. Sistem Pendukung Pengambilan Keputusan

Konsep Sistem Pendukung Keputusan (SPK) atau Decision Support System (DSS) pertama kali diperkenalkan oleh Michael S. Scott Morton pada awal tahun 1970-an. Konsep SPK ditandai dengan sistem interaktif berbasis komputer yang membantu pengambilan keputusan dengan memanfaatkan data dan model untuk menyelesaikan masalah yang bersifat tidak terstruktur dan semi terstruktur. (Turban, 2001).

Sistem pendukung keputusan didefinisikan sebagai sistem berbasis komputer yang terdiri dari komponen-komponen yang saling berinteraksi, yaitu: sistem bahasa, sistem pengetahuan, dan sistem pemrosesan masalah (Turban, 2010).

Sistem Pendukung Keputusan (SPK) bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dilengkapi dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. SPK ditujukan untuk membantu para pengambil keputusan untuk memecahkan masalah semi dan atau tidak terstruktur dengan fokus menyajikan informasi yang nantinya bisa dijadikan sebagai bahan alternatif pengambilan keputusan yang terbaik.

### B. Metode TOPSIS

#### 1. Konsep Dasar TOPSIS

TOPSIS berakar pada ide bahwa alternatif terbaik bukan hanya yang memiliki jarak terpendek dari solusi ideal positif, tetapi juga memiliki jarak terpanjang dari solusi ideal negatif. Keunggulan TOPSIS sering digunakan karena konsepnya yang sederhana dan mudah dimengerti, efisiensi komputasinya, dan kemampuannya untuk mengukur kinerja relatif dari alternatif keputusan secara matematis yang sederhana. Metode ini mengevaluasi jarak suatu alternatif terhadap solusi ideal positif dan solusi ideal negatif secara simultan. Penentuan solusi optimal dalam TOPSIS melibatkan evaluasi kedekatan relatif suatu alternatif terhadap solusi ideal positif, yang kemudian digunakan untuk merangking alternatif

berdasarkan prioritas nilai kedekatan tersebut. Hasil peringkat ini dapat menjadi pedoman bagi pengambil keputusan dalam memilih solusi terbaik.

## 2. Kelebihan dan Kekurangan Metode TOPSIS

Dalam metode TOPSIS terdapat beberapa kelebihan dan kekurangan diantaranya adalah sebagai berikut:

### a) Kelebihan

- Konsepnya yang sederhana dan mudah dipahami.
- Komputasi efisien.
- Dapat menentukan nilai preferensi untuk setiap alternatif dari alternatif – alternatif keputusan dalam bentuk matematis yang sederhana.

### b) Kekurangan

- Belum adanya penentuan bobot prioritas yang menjadi hitungan terhadap kriteria yang berguna untuk meningkatkan validitas nilai bobot perhitungan kriteria.
- Belum adanya mediator seperti hierarki jika di proses secara mandiri, maka dalam ketepatan pengambilan keputusan cenderung belum menghasilkan keputusan yang sempurna.

## 3. Langkah Penyelesaian Metode TOPSIS

- a) Normalisasi nilai atribut untuk membentuk matriks ternormalisasi (R) dapat menggunakan persamaan.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad \dots (1)$$

dengan  $i = 1, 2, 3, \dots, n$

dan  $j = 1, 2, 3, \dots, m$

- b) Membuat matriks keputusan yang ternormalisasi terbobot (Y). Perkalian antara bobot dengan nilai setiap atribut dihitung menggunakan persamaan

$$y_{ij} = w_i r_{ij} \quad \dots (2)$$

- c) Menentukan matriks solusi ideal positif (A+) dan matriks solusi ideal negatif (A-) dapat menggunakan persamaan

$$A^+ = y_1^+, y_2^+, \dots, y_j^+ \quad \dots (3)$$

$$A^- = y_1^-, y_2^-, \dots, y_j^-$$

di mana :

$$y_1^+ \begin{cases} \text{Max}_i y_{ij}; \text{ jika } j \text{ adalah atribut benefit} \\ \text{Min}_i y_{ij}; \text{ jika } j \text{ adalah atribut cost} \end{cases}$$

$$y_1^- \begin{cases} \text{Min}_i y_{ij}; \text{ jika } j \text{ adalah atribut benefit} \\ \text{Max}_i y_{ij}; \text{ jika } j \text{ adalah atribut cost} \end{cases}$$

- d) Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif.

- Jarak antara alternatif dengan solusi ideal positif dapat menggunakan persamaan

$$D_i^+ = \sqrt{\sum_{j=1}^m (y_i^+ - y_{ij})^2} \quad \dots (4)$$

- Jarak antara alternatif dengan solusi ideal negatif dapat menggunakan persamaan

$$D_i^- = \sqrt{\sum_{j=1}^m (y_{ij} - y_j^-)^2} \dots (5)$$

- e) Menentukan nilai preferensi untuk setiap alternatif. Nilai preferensi merupakan kedekatan suatu alternatif terhadap solusi ideal. Nilai preferensi untuk setiap alternatif ( $V_i$ ) dapat menggunakan persamaan

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \dots (6)$$

### III. HASIL DAN PEMBAHASAN

#### A. Data Kriteria

Peneliti menggunakan beberapa kriteria dalam hal menentukan *ranking* pemilihan bintang tamu untuk konser di Yogyakarta. Adapun kriteria sebagai berikut :

##### 1. Popularitas

Jumlah pendengar setiap bulannya yang tercatat oleh aplikasi *Spotify*.

**Tabel 1. Pembobotan Kriteria Popularitas**

<i>Popularitas</i>	<i>Bobot</i>
< 1.000.000	1
1.000.000 – 5.000.000	2
5.000.001 – 8.000.000	3
8.000.001 – 10.000.000	4
> 10.000.000	5

##### 2. Harga

Biaya pokok untuk mengundang penyanyi atau *band*.

**Tabel 2. Pembobotan Kriteria Harga**

<i>Harga</i>	<i>Bobot</i>
--------------	--------------

> Rp 200.000.000,00	1
Rp 100.000.001,00 – Rp 200.000.000,00	2
Rp 50.000.001,00 – Rp 100.000.000,00	3
Rp 25.000.001,00 – Rp 50.000.000,00	4
≤ Rp 25.000.000,00	5

##### 3. Lokasi

Pusat aktivitas atau lokasi tempat tinggal penyanyi atau *band*.

**Tabel 3. Pembobotan Kriteria Lokasi**

<i>Lokasi</i>	<i>Bobot</i>
Di Luar Pulau Jawa dan Bali	1
Bali	2
Jakarta	3
Jawa Barat, Jawa Timur	4
Yogyakarta	5

Setelah menentukan kriteria penilaian, akan menentukan nilai bobot preferensi dari tiap-tiap kriteria berdasarkan tingkat kepentingan antara kriteria yang satu dengan yang lainnya. Nilai perbandingan tingkat kepentingan antara kriteria dapat dinyatakan dengan pernyataan sebagai berikut :

**Tabel 4. Standar Nilai Bobot**

<i>Nilai</i>	<i>Keterangan</i>
1	Tidak Penting
2	Kurang Penting
3	Cukup Penting
4	Penting
5	Sangat Penting

Dari yang ditunjukkan pada Tabel 4, diketahui bobot preferensi berdasarkan kriteria adalah seperti pada Tabel 5.

**Tabel 5. Standar Bobot Preferensi Kriteria**

<i>Kriteria</i>	<i>Bobot</i>	<i>Cost/Benefit</i>
Popularitas (C1)	4	Benefit
Harga (C2)	5	Cost
Lokasi (C3)	4	Benefit

#### B. Data Penyanyi dan Band

Terdapat 6 penyanyi dan band Indonesia yang dijadikan sebagai alternatif dalam penelitian pemilihan bintang tamu konser berdasarkan kriteria yang telah ditentukan seperti yang tertera pada Tabel 6.

**Tabel 6. Data Alternatif**

<i>Kode</i>	<i>Penyanyi / Band</i>
A1	Yovie & Nuno
A2	Juicy Luicy
A3	Tulus
A4	Tiara Andini
A5	NDX A.K.A
A6	Fiersa Besari

Setelah diketahui standar terhadap nilai bobot, bobot preferensi, dan data alternatif, dilanjutkan dengan kecocokan alternatif terhadap kriteria-kriteria yang sudah ada. Kecocokan alternatif terhadap kriteria didapatkan dari artikel-artikel yang membantu memberi gambaran perkiraan untuk harga *guest star*, statistika pendengar setiap bulan-nya pada aplikasi *Spotify*, dan wawancara langsung oleh peneliti. Dibawah ini adalah table kecocokan terhadap kriteria yang ditunjukkan pada Tabel 7.

**Tabel 7. Kecocokan Alternatif Terhadap Kriteria**

<i>Kode</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>
A1	3	1	4
A2	2	3	4
A3	4	2	4
A4	4	3	3
A5	3	3	5
A6	3	5	4

#### C. Perhitungan Metode TOPSIS

##### 1. Menentukan Matriks Keputusan Normalisasi

$$X1 = \sqrt{3^2 + 2^2 + 4^2 + 4^2 + 3^2 + 3^2} = 7,9373$$

$$r11 = \frac{3}{7,9373} = 0,378$$

$$r21 = \frac{2}{7,9373} = 0,252$$

$$r31 = \frac{4}{7,9373} = 0,504$$

$$r41 = \frac{4}{7,9373} = 0,504$$

$$r51 = \frac{3}{7,9373} = 0,378$$

$$r61 = \frac{3}{7,9373} = 0,378$$

$$X2 = \sqrt{1^2 + 3^2 + 2^2 + 3^2 + 3^2 + 5^2} = 7,5498$$

$$r12 = \frac{1}{7,5498} = 0,1325$$

$$r22 = \frac{3}{7,5498} = 0,3974$$

$$r32 = \frac{2}{7,5498} = 0,2649$$

$$r42 = \frac{3}{7,5498} = 0,3974$$

$$r52 = \frac{3}{7,5498} = 0,3974$$

$$r62 = \frac{5}{7,5498} = 0,6623$$

$$X3 = \sqrt{4^2 + 4^2 + 4^2 + 3^2 + 5^2 + 4^2} = 9,8995$$

$$r13 = \frac{4}{9,8995} = 0,4041$$

$$r23 = \frac{4}{9,8995} = 0,4041$$

$$r33 = \frac{4}{9,8995} = 0,4041$$

$$r43 = \frac{3}{9,8995} = 0,3030$$

$$r53 = \frac{5}{9,8995} = 0,5051$$

$$r63 = \frac{4}{7,5498} = 0,4041$$

Sehingga diperoleh nilai (R) sebagai berikut :

$$R = \begin{bmatrix} 0,3780 & 0,1325 & 0,4041 \\ 0,2520 & 0,3974 & 0,4041 \\ 0,5040 & 0,2649 & 0,4041 \\ 0,5040 & 0,3974 & 0,3030 \\ 0,3780 & 0,3974 & 0,5051 \\ 0,3780 & 0,6623 & 0,4041 \end{bmatrix}$$

2. *Menentukan Matriks Keputusan Ternormalisasi Terbobot*  
 $W = (4, 5, 4)$

$$\begin{aligned} y_{11} &= 4 \times 0,378 = 1,5119 \\ y_{21} &= 4 \times 0,252 = 1,0079 \\ y_{31} &= 4 \times 0,504 = 2,0158 \\ y_{41} &= 4 \times 0,504 = 2,0158 \\ y_{51} &= 4 \times 0,378 = 1,5119 \\ y_{61} &= 4 \times 0,378 = 1,5119 \end{aligned}$$

$$\begin{aligned} y_{12} &= 5 \times 0,1325 = 0,6623 \\ y_{22} &= 5 \times 0,3974 = 1,9868 \\ y_{32} &= 5 \times 0,2649 = 1,3245 \\ y_{42} &= 5 \times 0,3974 = 1,9868 \\ y_{52} &= 5 \times 0,3974 = 1,9868 \\ y_{62} &= 5 \times 0,6623 = 3,3113 \end{aligned}$$

$$\begin{aligned} y_{13} &= 4 \times 0,4041 = 1,6162 \\ y_{23} &= 4 \times 0,4041 = 1,6162 \\ y_{33} &= 4 \times 0,4041 = 1,6162 \\ y_{43} &= 4 \times 0,3030 = 1,2122 \\ y_{53} &= 4 \times 0,5051 = 2,0203 \\ y_{63} &= 4 \times 0,4041 = 1,6162 \end{aligned}$$

Sehingga diperoleh matriks Y :

$$Y = \begin{bmatrix} 1,5119 & 0,6623 & 1,6162 \\ 1,0079 & 1,9868 & 1,6162 \\ 2,0158 & 1,3245 & 1,6162 \\ 2,0158 & 1,9868 & 1,2122 \\ 1,5119 & 1,9868 & 2,0203 \\ 1,5119 & 3,3113 & 1,6162 \end{bmatrix}$$

3. *Menentukan Matriks Solusi Ideal Positif dan Solusi Ideal Negatif*

$$Y_1^+ = \max \{ 1,5119; 1,0079; 2,0158; 2,0158; 1,5119; 1,5119 \} = 2,0158$$

$$Y_2^+ = \max \{ 0,6623; 1,9868; 1,3245; 1,9868; 1,9868; 3,3113 \} = 0,6623$$

$$Y_3^+ = \max \{ 1,6162; 1,6162; 1,6162; 1,2122; 2,0203; 1,6162 \} = 2,0203$$

$$A^+ = \max \{ 2,0158; 0,6623; 2,0203 \}$$

$$Y_1^- = \max \{ 1,5119; 1,0079; 2,0158; 2,0158; 1,5119; 1,5119 \} = 1,0079$$

$$Y_2^- = \max \{ 0,6623; 1,9868; 1,3245; 1,9868; 1,9868; 3,3113 \} = 3,3113$$

$$Y_3^- = \max \{ 1,6162; 1,6162; 1,6162; 1,2122; 2,0203; 1,6162 \} = 1,2122$$

$$A^- = \max \{ 1,0079; 3,3113; 1,2122 \}$$

4. *Menentukan Jarak Antara Nilai Setiap Alternatif dengan Matriks Solusi Ideal Positif dan Matriks Solusi Ideal Negatif*

$$D_1^+ = \sqrt{(1,5119 - 2,0158)^2 + (0,6623 - 0,6623)^2 + (1,6162 - 2,0203)^2} = 0,6459$$

$$D_2^+ = \sqrt{(1,0079 - 2,0158)^2 + (1,9868 - 0,6623)^2 + (1,6162 - 2,0203)^2} = 1,7128$$

$$D_3^+ = \sqrt{(2,0158 - 2,0158)^2 + (1,3245 - 0,6623)^2 + (1,6162 - 2,0203)^2} = 0,7758$$

$$D_4^+ = \sqrt{(2,0158 - 2,0158)^2 + (1,9868 - 0,6623)^2 + (1,2122 - 2,0203)^2} = 1,5516$$

$$D_5^+ = \sqrt{(1,5119 - 2,0158)^2 + (1,9868 - 0,6623)^2 + (2,0203 - 2,0203)^2} = 1,4172$$

$$D_6^+ = \sqrt{(1,5119 - 2,0158)^2 + (3,3113 - 3,3113)^2 + (1,6162 - 1,2122)^2} = 2,7267$$

$$D_1^- = \sqrt{(1,5119 - 1,0079)^2 + (0,6623 - 3,3113)^2 + (1,6162 - 1,2122)^2} = 2,7267$$

$$\begin{aligned}
D_2^- &= \sqrt{(1,0079 - 1,0079)^2 + (1,9868 - 3,3113)^2 + (1,6162 - 1,2122)^2} \\
&= 1,3848 \\
D_3^- &= \sqrt{(2,0158 - 1,0079)^2 + (1,3245 - 3,3113)^2 + (1,6162 - 1,2122)^2} \\
&= 2,2642 \\
D_4^- &= \sqrt{(2,0158 - 1,0079)^2 + (1,9868 - 3,3113)^2 + (1,2122 - 1,2122)^2} \\
&= 1,6644 \\
D_5^- &= \sqrt{(1,5119 - 1,0079)^2 + (1,9868 - 3,3113)^2 + (2,0203 - 1,2122)^2} \\
&= 1,6314 \\
D_6^- &= \sqrt{(1,5119 - 1,0079)^2 + (3,3113 - 3,3113)^2 + (1,6162 - 1,2122)^2} \\
&= 0,6459
\end{aligned}$$

5. Menentukan Nilai Preferensi untuk Setiap Alternatif

$$\begin{aligned}
V_1 &= \frac{2,7267}{2,7267+0,6459} = 0,8085 \\
V_2 &= \frac{1,3848}{1,3848+1,7128} = 0,4471 \\
V_3 &= \frac{2,2642}{2,2642+0,7758} = 0,7448 \\
V_4 &= \frac{1,6644}{1,6644+1,5516} = 0,5175 \\
V_5 &= \frac{1,6314}{1,6314+1,4172} = 0,5351 \\
V_6 &= \frac{0,6459}{0,6459+2,7267} = 0,1915
\end{aligned}$$

**Tabel 8. Nilai Perangkingan Preferensi Setiap Alternatif**

Alternatif	Nilai Preferensi
A1	0,8085
A3	0,7448
A5	0,5351
A4	0,5175
A2	0,4471
A6	0,1915

Dari hasil perhitungan secara manual diatas, alternatif dengan kode A1 yaitu 'Yovie & Nuno' memiliki nilai preferensi tertinggi dengan nilai preferensi 0,8085. Dengan demikian, 'Yovie & Nuno' merupaksan opsi pertama atau opsi utama dalam pemilihan *guest star* konser di Yogyakarta.

#### IV. PENGUJIAN DAN ANALISIS

##### A. Instrumen Pengujian

Pada pengujian dan analisis ini digunakan untuk menguji sistem yang telah dibuat. Dalam pengujian ini dilakukan *Beta testing* untuk mengetahui kualitas sistem dan apakah sistem ini sudah sesuai dengan apa yang diinginkan oleh user. Pengujian ini dinilai oleh pengguna sistem menggunakan kuesioner dengan pertanyaan dan skala penilaian yang telah ditentukan oleh peneliti,

**Tabel 9. Instrumen Pengujian Beta**

No	Pertanyaan	Skala
1	Apakah SPPK tersebut memudahkan proses pemilihan <i>guest star</i> konser musik di Yogyakarta?	1-5
2	Apakah SPPK tersebut membantu memberikan informasi yang cukup dan relevan mengenai pemilihan <i>guest star</i> konser musik di Yogyakarta?	1-5
3	Apakah SPPK pemilihan <i>guest star</i> konser musik di Yogyakarta sudah memenuhi kebutuhan user?	1-5
4	Apakah user terbantu dengan adanya pemilihan <i>guest star</i> konser musik di Yogyakarta?	1-5
5	Apakah SPPK pemilihan <i>guest star</i> konser musik di Yogyakarta membantu	1-5

	meningkatkan efektivitas pekerjaan?	
6	Apakah SPPK ini dapat diandalkan dalam mendukung kegiatan pemilihan guest star konser musik?	1-5
7	Apakah SPPK ini dapat membantu dalam mengoptimalkan anggaran atau biaya terkait pemilihan guest star?	1-5

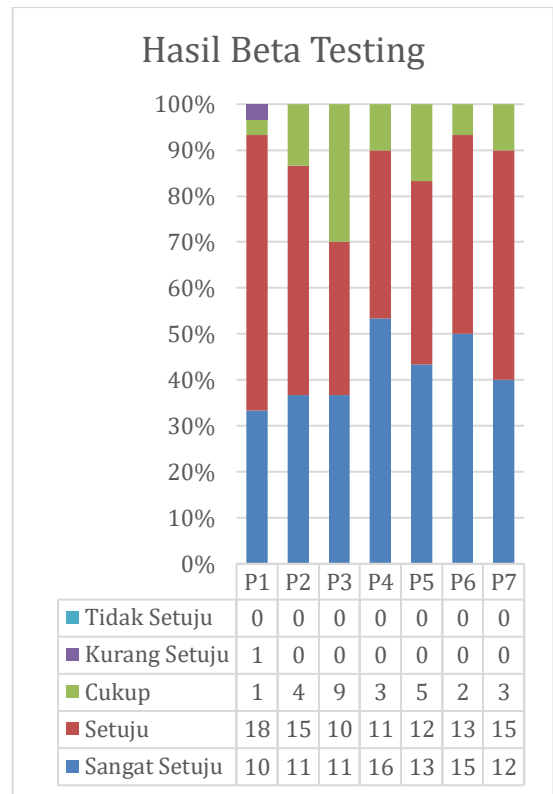
**Tabel 10. Skala Penilaian**

<i>Skala</i>	<i>Keterangan</i>
1	Tidak Setuju
2	Kurang Setuju
3	Cukup
4	Setuju
5	Sangat Setuju

#### B. Analisis Hasil Pengujian

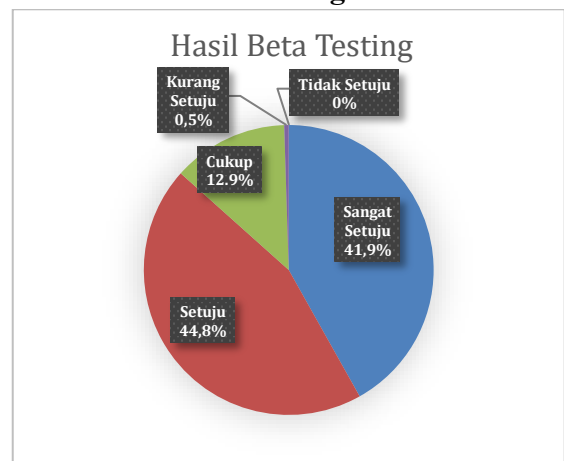
Pengujian ini melibatkan penggunaan kuesioner yang telah disebar kepada sejumlah responden. Kuesioner tersebut terdiri dari tujuh pertanyaan, masing-masing dengan lima pilihan jawaban. Setiap pilihan jawaban memiliki nilai yang telah ditentukan sebelumnya. Berikut adalah hasil dari pengujian yang telah dilakukan:

**Gambar 1. Diagram Batang**



Hasil pengujian beta, yang mencakup perhitungan kategori jawaban dari kuesioner yang telah didistribusikan, jelas tergambar dalam Gambar 1. Selanjutnya, hasil ini direpresentasikan dalam bentuk diagram *pie* pada Gambar 2.

**Gambar 2. Diagram Pie**



Dari Gambar 2, dapat diobservasi bahwa persentase keseluruhan pengujian beta menunjukkan bahwa 86,7% dari responden setuju, dengan 41,9% sangat



setuju dan 44,8% setuju. Dengan itu, dapat dinyatakan bahwa Sistem Pendukung Pengambilan Keputusan Pemilihan Guest Star pada konser musik di Yogyakarta, yang menggunakan metode TOPSIS tersebut efektif dan mampu memberikan bantuan kepada pengguna dalam menyajikan informasi mengenai opsi *guest star* dengan baik, mampu memberikan informasi yang bermanfaat terkait pemilihan *guest star* konser musik di Yogyakarta.

## V. KESIMPULAN DAN SARAN

Berdasarkan hasil perhitungan yang telah dilakukan, maka dapat disimpulkan bahwa dengan menggunakan metode TOPSIS, dapat membantu pihak pengambil keputusan dalam mengidentifikasi alternatif *guest star* yang akan dipilih berdasarkan hasil *ranking* tertinggi.

Hasil analisis dari perhitungan metode TOPSIS menyatakan bahwa alternatif yang terpilih sebagai rekomendasi utama adalah ‘Yovie & Nuno’ karena memiliki nilai preferensi tertinggi dari 3(tiga) aspek kriteria yang ditentukan, antara lain : popularitas, harga, dan lokasi.

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, terdapat beberapa saran yang perlu dipertimbangkan dalam pengembangan penelitian ini. Pertama, diperlukan penambahan kriteria yang mampu mengukur besarnya komunitas penggemar yang dimiliki oleh *guest star*. Selanjutnya, penting untuk mempertimbangkan jumlah orang dalam tim *guest star* (riders) sebagai kriteria penilaian. Terakhir, diharapkan agar hasil penelitian ini dapat diimplementasikan dalam bentuk sebuah website.

## REFERENSI

- [1] Nurhusni, K. S. (2019). *ANALISIS PERBANDINGAN METODE SAW, TOPSIS DAN SAW-TOPSIS UNTUK MENENTUKAN MEDIA SOSIAL TERBAIK SEBAGAI SARANA JUAL BELI ONLINE*.
- [2] Pujiastuti, A., Kusumaningrum, A., & Sofyan, M. A. (2020). *WEB-BASED DECISION SUPPORT SYSTEM FOR GUEST STAR DETERMINATION IN E-MARKETPLACE USING WEIGHTED PRODUCT METHOD*.  
<http://eprints.stta.ac.id/895/1/peer%20review%20web%20based.pdf>
- [3] Sugiarto, H. (2021). *Penerapan Metode Topsis Untuk Pemilihan Perumahan*, 7.  
<https://ejournal.bsi.ac.id/ejurnal/index.php/jtk/article/download/10411/pdf>
- [4] <https://dspace.uui.ac.id/bitstream/handle/123456789/3781/05.2%20bab%202.pdf?seque>

## LAMPIRAN

## LISTING CODE

Code	Penjelasan
<pre>alternatif = ["Yovie &amp; nuno", "Juicy Luicy", "Tulus", "Tiara Andini", "NDX A.K.A", "Fiersa Besari"]</pre>	<p>Melakukan deklarasi variable alternatif. Variable alternatif menyimpan isi dari nama pada data calon guest star. Kemudian mencetak isi dari variable alternatif dengan menggunakan str agar list menjadi string sebelum dicetak.</p>
<pre>kriteria = ["Popularitas", "Harga", "Lokasi" ] costbenefit = ["benefit", "cost", "benefit"] kepentingan = [4, 5, 4] alternatifkriteria=[ [3, 1, 4], [2, 3, 4], [4, 2, 4], [4, 3, 3], [3, 3, 5], [3, 5, 4]]</pre>	<p>Mendeklarasikan 4 variabel yang akan digunakan. Variable kriteria berisi 3 kriteria yang digunakan untuk melakukan seleksi.</p> <p>Variable costbenefit berisi informasi untuk kriteria. Disetiap kriteria akan diberikan petunjuk termasuk kriteria biaya(cost) atau kriteria keuntungan (benefit).</p> <p>Variable kepentingan berisi bobot untuk setiap kriteria.</p> <p>Variable alternatifkriteria berisi penilaian alternatif terhadap setiap kriteria yang ada pada. Terdapat 6 alternatif dengan 3 kriteria.</p>
<pre>#Mencetak isi variabel alternatif print(str(alternatif))</pre>	<p>Mencetak isi dari variable alternatif yang berisi 6 nama yang ada pada data alternatif.</p>
<pre>#Mencetak isi variabel kriteria print(str(kriteria))</pre>	<p>Mencetak isi dari variable kriteria yang berisi 3 kriteria yang akan digunakan untuk menganalisa.</p>
<pre>#Mencetak isi variabel costbenefit print(str(costbenefit))</pre>	<p>Mencetak isi dari variable costbenefit yang berisi informasi kriteria tersebut masuk pada cost atau benefit</p>
<pre>#Mencetak isi variabel alternatif kriteria print(str(alternatifkriteria))</pre>	<p>Mencetak isi dari variable alternatifkriteria yang berisi nilai alternatif pada kasus penerima bantuan rumah layak huni.</p>
<pre>#Code untuk mendapatkan nilai alternatif pembagi = [] for i in range(len(kriteria)):     pembagi.append(0)     for j in range(len(alternatif)):         pembagi[i] = pembagi[i] + (alternatifkriteria[j][i] * alternatifkriteria[j][i])         pembagi[i]= pembagi[i]**(1/2.0) print(str(pembagi))</pre>	<p>Mendeklarasikan variable pembagi dengan list kosong untuk menyimpan nilai pembagi pada setiap kriteria. Menggunakan looping I untuk mengiterasi melalui setiap kriteria yang kemudian dengan pembagi.append(0) akan menambahkan elemen awal 0 ke list pembagi. Melakukan loop j untuk mengiterasi melalui setiap alternatif. Dimasukkan rumus untuk mendapatkan nilai akar kuadrat dari penjumlahan kuadrat yang kemudian hasil perhitungan dicetak untuk setiap kriteria.</p>
<pre>#Code untuk mendapatkan nilai matriks ternormalisasi normalisasi = [] for i in range(len(alternatif)):     normalisasi.append([])     for j in range(len(kriteria)):</pre>	<p>Mendeklarasikan variable normalisasi dengan isi list kosong yang akan digunakan untuk menyimpan matriks data yang telah ternormalisasi. Pada loop I digunakan untuk mengiterasi melalui setiap alternatif yang kemudian hasil akan disimpan ke normalisasi.append untuk membuat daftar nilai</p>

<pre>         normalisasi[i].append(0)         normalisasi[i][j] = alternatifkriteria[i][j] / pembagi[j] print(str(normalisasi)) </pre>	<p>normalisasi untuk alternatif. Loop j untuk mengiterasi melalui setiap kriteria pada setiap alternatif. Menggunakan rumus alternatifkriteria / pembagi kemudian disimpan ke variable normalisasi. Cetak hasil perhitungan yang dimana berisi matriks yang sudah dinormalisasi.</p>
<pre> #Code untuk mendapatkan matriks normalisasi terbobot terbobot = [] for i in range(len(alternatif)):     terbobot.append([])     for j in range(len(kriteria)):         terbobot[i].append(0)         terbobot[i][j] = normalisasi[i][j] * kepentingan[j] print(str(terbobot)) </pre>	<p>Mendeklarasikan variable terbobot dengan isi list kosong yang akan digunakan untuk menyimpan matriks terbobot. Pada loop I digunakan untuk mengiterasi melalui setiap alternatif yang kemudian hasil akan disimpan ke terbobot.append untuk menambahkan elemen awal 0 kedalam list terbobot untuk setiap kriteria dan alternatif. Loop j untuk mengiterasi melalui setiap kriteria pada setiap alternatif. Menggunakan rumus normalisasi x kepentingan yang kemudian hasil disimpan ke variable terbobot. Cetak hasil dari perhitungan matriks terbobot.</p>
<pre> #Code untuk memnentukan solusi ideal positif aplus = [] for i in range(len(kriteria)):     aplus.append(0)     if costbenefit[i]== 'cost':         for j in range(len(alternatif)):             if j == 0:                 aplus[i] = terbobot[j][i]             else:                 if aplus[i] &gt; terbobot[j][i]:                     aplus[i] = terbobot[j][i]             else: #costbenefit[i] == 'benefit':                 for j in range(len(alternatif)):                     if j == 0:                         aplus[i] = terbobot[j][i]                     else:                         if aplus[i] &lt; terbobot[j][i]:                             aplus[i] = terbobot[j][i] print(str(aplus)) </pre>	<p>Mendeklarasikan variable terbobot dengan isi list kosong yang akan digunakan untuk menyimpan nilai solusi ideal positif. Diberikan kondisi jika informasi kriteria pada costbenefit bernilai cost, akan dilakukan pemeriksaan nilai terbobot setiap alternatif pada kriteria untuk mendapatkan cost yang berisi nilai terendah dari seluruh alternatif. Jika kondisi informasi kriteria benefit, akan dilakukan pemeriksaan nilai terbobot setiap alternatif pada setiap kriteria untuk mendapatkan benefit yang berisi nilai tertinggi dari seluruh alternatif. Kemudian hasil dicetak yang isinya berupa list yang berisi nilai A+ untuk setiap kriteria.</p>
<pre> #Code untuk mendapatkan solusi ideal negatif amin = [] for i in range(len(kriteria)):     amin.append(0)     if costbenefit[i]== 'cost':         for j in range(len(alternatif)): </pre>	<p>Mendeklarasikan variable amin dengan isi list kosong yang akan digunakan untuk menyimpan nilai solusi ideal negatif. Diberikan kondisi jika informasi kriteria pada costbenefit bernilai cost, akan dilakukan pemeriksaan nilai terbobot setiap alternatif pada kriteria untuk mendapatkan cost yang berisi nilai tertinggi dari seluruh alternatif.</p>

<pre>         if j == 0:             amin[i] =             terbobot[j][i]         else:             if amin[i] &lt;             terbobot[j][i]:                 amin[i] =                 terbobot[j][i]             else:                 for j in                 range(len(alternatif)):                     if j == 0:                         amin[i] =                         terbobot[j][i]                     else:                         if amin[i] &gt;                         terbobot[j][i]:                             amin[i] =                             terbobot[j][i] print(str(amin)) </pre>	<p>Jika kondisi informasi kriteria benefit, akan dilakukan pemeriksaan nilai terbobot setiap alternatif pada setiap kriteria untuk mendapatkan benefit yang berisi nilai terendah dari seluruh alternatif. Kemudian hasil dicetak yang isinya berupa list yang berisi nilai A- untuk setiap kriteria.</p>
<pre> #code untuk mendapatkan jarak alternatif dgn solusi ideal positif dplus=[] for i in range(len(alternatif)):     dplus.append(0)     for j in     range(len(kriteria)):         dplus[i] = dplus[i] +         ((aplus[j] - terbobot[i][j]) *         (aplus[j] - terbobot[i][j]))         dplus[i]= dplus[i]**(1/2.0) print(str(dplus)) </pre>	<p>Mendeklarasikan variable dplus dengan isi list kosong yang akan digunakan untuk menyimpan nilai jarak antara setiap alternatif dengan solusi ideal positif. Pada loop I digunakan untuk mengiterasi melalui setiap alternatif yang kemudian hasil akan disimpan ke dplus.append untuk menambahkan elemen awal 0 kedalam list terbobot untuk setiap kriteria dan alternatif. Loop j untuk mengiterasi melalui setiap kriteria pada setiap alternatif. Untuk mendapatkan hasil, dilakukan perhitungan akar kuadrat dari nilai jarak. Hasil dicetak yang pada hasilnya berisi hasil perhitungan jarak antara setiap alternatif dengan A+</p>
<pre> #Code untuk mendapatkan jarak alternatif dengan solusi ideal negatif dmin=[] for i in range(len(alternatif)):     dmin.append(0)     for j in     range(len(kriteria)):         dmin[i] = dmin[i] +         ((terbobot[i][j] - amin[j]) *         (terbobot[i][j] - amin[j]))         dmin[i]= dmin[i]**(1/2.0) print(str(dmin)) </pre>	<p>Mendeklarasikan variable dmin dengan isi list kosong yang akan digunakan untuk menyimpan nilai jarak antara setiap alternatif dengan solusi ideal negatif. Pada loop I digunakan untuk mengiterasi melalui setiap alternatif yang kemudian hasil akan disimpan ke dmin.append untuk menambahkan elemen awal 0 kedalam list terbobot untuk setiap kriteria dan alternatif. Loop j untuk mengiterasi melalui setiap kriteria pada setiap alternatif. Untuk mendapatkan hasil, dilakukan perhitungan akar kuadrat dari nilai jarak. Hasil dicetak yang pada hasilnya berisi hasil perhitungan jarak antara setiap alternatif dengan A-</p>
<pre> #code untuk menyimpan nilai preferensi untuk setiap alternatif hasil = [] for i in range(len(alternatif)): </pre>	<p>Mendeklarasikan variable hasil dengan isi list kosong yang akan digunakan untuk menyimpan nilai kedekatan untuk setiap alternatif. Kemudian menggunakan rumus dmin : (dmin + dplus) kemudian mencetak variable alternatif.</p>

<pre> hasil.append(0) for j in range(len(kriteria)):     hasil[i] = dmin[i] / (dmin[i] + dplus[i]) print(str(alternatif)) print(str(hasil)) </pre>	
<pre> alternatifrangking = [] </pre>	Mencetak isi variable hasil
<pre> hasilrangking = [] </pre>	Mendeklarasikan variable alternatifrangking
<pre> for i in range(len(alternatif)):     hasilrangking.append(hasil[i])     alternatifrangking.append(alternatif[i]) </pre>	Mendeklarasikan variabel hasilrangking
<pre> for i in range(len(alternatif)):     for j in range(len(alternatif)):         if j &gt; i:             if hasilrangking[j] &gt; hasilrangking[i]:                 tmphasil = hasilrangking[i]                 tmpalternatif = alternatifrangking[i]                 hasilrangking[i] = hasilrangking[j]                 alternatifrangking[i] = alternatifrangking[j]                 hasilrangking[j] = tmphasil                 alternatifrangking[j] =tmpalternatif </pre>	Melakukan iterasi ke semua alternatif dan memindahkan hasil ke hasilrangking dan memindahkan alternatif ke alternatifrangking
<pre> for i in range(len(alternatif)):     for j in range(len(alternatif)):         if j &gt; i:             if hasilrangking[j] &gt; hasilrangking[i]:                 tmphasil = hasilrangking[i]                 tmpalternatif = alternatifrangking[i]                 hasilrangking[i] = hasilrangking[j]                 alternatifrangking[i] = alternatifrangking[j]                 hasilrangking[j] = tmphasil                 alternatifrangking[j] =tmpalternatif </pre>	Memilih satu alternatif untuk menjadi referensi kemudian membandingkan skor kedekatan alternatif lain dengan alternatif referensi.
<pre> print(str(alternatifrangking)) </pre>	Mencetak variable alternatifrangking
<pre> print(str(hasilrangking)) </pre>	Mencetak variable hasil rangking

## PROGRAM

```
+ Code + Text
Connect ^

[ ] alternatif = ["Yovie & nuno", "Juicy Luicy", "Tulus", "Tiara Andini", "NDX A.K.A", "Fiersa Besari"]

[ ] kriteria = ["Popularitas", "Harga", "Lokasi"]
costbenefit = ["benefit", "cost", "benefit"]
kepinganan = [4,5,4]
alternatifkriteria=[[3,1,4],
[2,3,4],
[4,2,4],
[4,3,3],
[3,3,5],
[3,5,4]]

[ ] #Mencetak isi variabel alternatif
print(str(alternatif))

['Yovie & nuno', 'Juicy Luicy', 'Tulus', 'Tiara Andini', 'NDX A.K.A', 'Fiersa Besari']

[ ] #Mencetak isi variabel kriteria
print(str(kriteria))

['Popularitas', 'Harga', 'Lokasi']

[ ] #Mencetak isi variabel costbenefit
print(str(costbenefit))

['benefit', 'cost', 'benefit']

[ ] #Mencetak isi variabel alternatif kriteria
print(str(alternatifkriteria))

[[3, 1, 4], [2, 3, 4], [4, 2, 4], [4, 3, 3], [3, 3, 5], [3, 5, 4]]

Metode Technique for Order of Preference by Similarity

[ ] #Code untuk mendapatkan nilai alternatif
pembagi = []
for i in range(len(kriteria)):
    pembagi.append(0)
    for j in range(len(alternatif)):
        pembagi[i] = pembagi[i] + (alternatifkriteria[j][i] * alternatifkriteria[j][i])
    pembagi[i] = pembagi[i]**(1/2.0)
    print(str(pembagi))

[7.937253933193772, 7.54983443527075, 9.899494936611665]

[ ] #Code untuk mencapata nilai matriks ternormalisasi
normalisasi = []
for i in range(len(alternatif)):
    normalisasi.append([])
    for j in range(len(kriteria)):
        normalisasi[i].append(0)
        normalisasi[i][j] = alternatifkriteria[i][j] / pembagi[j]
    print(str(normalisasi))

[[0.3779644730092272, 0.13245323570650439, 0.40406101782088427], [0.2519763153394848, 0.39735970711951313, 0.40406101782088427], [0.5039526306789696, 0.26490647141300877, 0.40406101782088427]]

R Ternormalisasi

[ ] #Code untuk mendapatkan matriks normalisasi terbobot
terbobot = []
for i in range(len(alternatif)):
    terbobot.append([])
    for j in range(len(kriteria)):
        terbobot[i].append(0)
        terbobot[i][j] = normalisasi[i][j] * kepinganan[j]
    print(str(terbobot))

[[1.5118578920369088, 0.6622661785325219, 1.616244071283537], [1.0079052613579391, 1.9867985355975657, 1.616244071283537], [2.0158105227158782, 1.3245323570650438, 1.616244071283537]]

Solusi Positif

[ ] #Code untuk memnentukan solusi ideal positif
aplus = []
for i in range(len(kriteria)):
    apus.append(0)
    if costbenefit[i] == 'cost':
        for j in range(len(alternatif)):
            if j == 0:
                apus[i] = terbobot[j][i]
            else:
                if apus[i] > terbobot[j][i]:
                    apus[i] = terbobot[j][i]
    else: #costbenefit[i] == 'benefit':
        for j in range(len(alternatif)):
            if j == 0:
                apus[i] = terbobot[j][i]
            else:
                if apus[i] < terbobot[j][i]:
                    apus[i] = terbobot[j][i]
    print(str(apus))

[2.0158105227158782, 0.6622661785325219, 2.0203050891044216]
```

```
[ ] #Code untuk mendapatkan solusi ideal negatif
amin = []
for i in range(len(kriteria)):
    amin.append(0)
    if costbenefit[i] == 'cost':
        for j in range(len(alternatif)):
            if j == 0:
                amin[i] = terbotot[j][i]
            else:
                if amin[i] < terbotot[j][i]:
                    amin[i] = terbotot[j][i]
        else:
            for j in range(len(alternatif)):
                if j == 0:
                    amin[i] = terbotot[j][i]
                else:
                    if amin[i] > terbotot[j][i]:
                        amin[i] = terbotot[j][i]
print(str(amin))
```

```
[1.0079052613579391, 3.3113308926626095, 1.212183053462653]
```

```
[ ] #code untuk mendapatkan jarak alternatif dgn solusi ideal positif
dplus = []
for i in range(len(alternatif)):
    dplus.append(0)
    for j in range(len(kriteria)):
        dplus[i] = dplus[i] + ((apulus[j] - terbotot[i][j]) * (apulus[j] - terbotot[i][j]))
    dplus[i] = dplus[i]**(1/2.0)
print(str(dplus))
```

```
[0.6459361888690732, 1.7127534226816612, 0.7757975234238115, 1.5515950468476227, 1.4171641467665397, 2.7266788259235493]
```

```
[ ] #Code untuk mendapatkan jarak alternatif dengan solusi ideal negatif
dmin = []
for i in range(len(alternatif)):
    dmin.append(0)
    for j in range(len(kriteria)):
        dmin[i] = dmin[i] + ((terbotot[i][j] - amin[j]) * (terbotot[i][j] - amin[j]))
    dmin[i] = dmin[i]**(1/2.0)
print(str(dmin))
```

```
[2.7266788259235493, 1.384792862140302, 2.2641790439468554, 1.664409499127332, 1.631384517325799, 0.6459361888690731]
```

```
[ ] #code untuk menyimpan nilai preferensi untuk setiap alternatif
hasil = []
for i in range(len(alternatif)):
    hasil.append(0)
    for j in range(len(kriteria)):
        hasil[i] = dmin[i] / (dmin[i] + dplus[i])
    print(str(alternatif))
```

```
['Yovie & nuno', 'Juicy Luicy', 'Tulus', 'Tiara Andini', 'NDX A.K.A', 'Fiersa Besari']
```

```
[ ] print(str(hasil))
```

```
[0.8084761569180197, 0.4470612332496253, 0.7448014791460009, 0.5175395355738698, 0.5351348123588245, 0.19152384308198037]
```

```
[ ] alternatiffrangking = []
hasilrangking = []
```

```
[ ] for i in range(len(alternatif)):
    hasilrangking.append(hasil[i])
    alternatiffrangking.append(alternatif[i])
```

```
[ ] for i in range(len(alternatif)):
    for j in range(len(alternatif)):
        if j > i:
            if hasilrangking[j] > hasilrangking[i]:
                tmphasil = hasilrangking[i]
                tmpalternatif = alternatiffrangking[i]
                hasilrangking[i] = hasilrangking[j]
                alternatiffrangking[i] = alternatiffrangking[j]
                hasilrangking[j] = tmphasil
                alternatiffrangking[j] = tmpalternatif
```

```
[ ] print(str(alternatiffrangking))
```

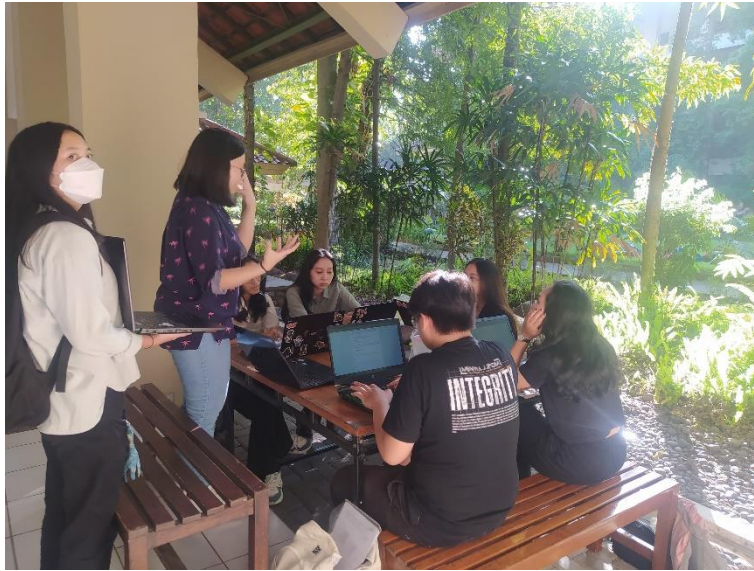
```
['Yovie & nuno', 'Tulus', 'NDX A.K.A', 'Tiara Andini', 'Juicy Luicy', 'Fiersa Besari']
```

```
[ ] print(str(hasilrangking))
```

```
[0.8084761569180197, 0.7448014791460009, 0.5351348123588245, 0.5175395355738698, 0.4470612332496253, 0.19152384308198037]
```



## DOKUMENTASI





## LAPORAN KERJA KELOMPOK

Posisi	Nama	Tugas
Koordinator Tim	Resiana Kinanti Jati (215314159)	Metodologi Penelitian, menyusun kuesioner
Anggota Tim	Marta Yulianti (215314154)	Pendahuluan, mengumpulkan data alternatif
Anggota Tim	Loadtriani Oktavia S (215314172)	Pemrograman, perhitungan dengan excel
Anggota Tim	Victoria Alysha F S (215314158)	Abstrak, penjelasan perhitungan, analisa hasil pengujian