

**APLIKASI METODE TOPSIS PADA PYTHON UNTUK MENENTUKAN
MOST VALUABLE PLAYER TURNAMEN MOBILE LEGENDS
PROFESSIONAL LEAGUE INDONESIA SEASON 5**

SKRIPSI

**OLEH
MOHAMMAD RIFQI RIZA PAHLEVY
NIM. 16610094**



**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**APLIKASI METODE TOPSIS PADA PYTHON UNTUK MENENTUKAN
MOST VALUABLE PLAYER TURNAMEN MOBILE LEGENDS
PROFESSIONAL LEAGUE INDONESIA SEASON 5**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**OLEH
MOHAMMAD RIFQI RIZA PAHLEVY
NIM. 16610094**

**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2020**

**APLIKASI METODE TOPSIS PADA PYTHON UNTUK MENENTUKAN
MOST VALUABLE PLAYER TURNAMEN MOBILE LEGENDS
PROFESSIONAL LEAGUE INDONESIA SEASON 5**

SKRIPSI

Oleh

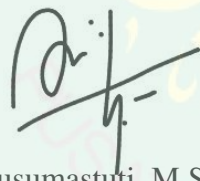
Mohammad Rifqi Riza Pahlevy

NIM. 16610094

Telah Diperiksa dan Disetujui untuk Diuji

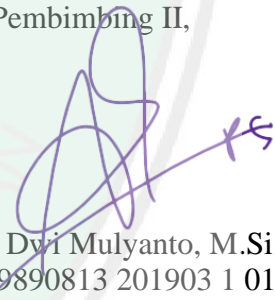
Tanggal 17 Juni 2020

Pembimbing I,



Ari Kusumastuti, M.Si, M.Pd
NIP. 19770521 200501 2 004

Pembimbing II,



Angga Dwi Mulyanto, M.Si
NIP. 19890813 201903 1 012

Mengetahui,

Ketua Jurusan Matematika



Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

**APLIKASI METODE TOPSIS PADA PYTHON UNTUK MENENTUKAN
MOST VALUABLE PLAYER TURNAMEN MOBILE LEGENDS
PROFESSIONAL LEAGUE INDONESIA SEASON 5**

SKRIPSI

Oleh

Mohammad Rifqi Riza Pahlevy

NIM. 16610094

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)


Tanggal 17 Juni 2020

Penguji Utama : Dr. H. Imam Sujarwo, M.Pd

Ketua Penguji : Abdul Aziz, M.Si

Sekretaris Penguji : Ari Kusumastuti, M.Si, M.Pd

Anggota Penguji : Angga Dwi Mulyanto, M.Si



Mengetahui,

Ketua Jurusan Matematika



Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Mohammad Rifqi Riza Pahlevy

NIM : 16610094

Jurusan : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Aplikasi Metode TOPSIS Pada Python Untuk Menentukan
Most Valuable Player Turnamen *Mobile Legends Professional League Indonesia Season 5*.

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 17 Juni 2020
Yang membuat pernyataan,



Mohammad Rifqi Riza Pahlevy
NIM. 16610094

MOTO

“Bahagiakan terlebih dahulu keluarga, In Sya Allah kamu akan bahagia bersama semua orang.”



PERSEMBAHAN

Penulis persembahkan skripsi ini kepada:

Ibunda Sunarti dan Ayahanda Akhmad Khusaeri yang selalu merawat, mendidik dan menyayangi anak-anaknya dari lahir hingga mengantarkan anaknya menuju jenjang perguruan tinggi. Kakak tersayang Alfin Riza Masyita dan Adik tercinta Haidar Daniel Muhammad Irsyad yang selalu mendukung setiap langkah yang ditempuh penulis. Serta keluarga besar Bani Abdul Chamid dan Bani H. Masyhud yang selalu menemani dalam lika-liku hidup penulis.

KATA PENGANTAR

Assalmu'alaikum Warahmatullahi Wabarakatuh

Segala puji syukur penulis panjatkan kehadirat Allah SWT yang selalu melimpahkan rahmat, taufik, serta hidayah-Nya. Shalawat serta salam senantiasa tercurahkan kepada Nabi Muhaammad SAW yang telah menuntun manusia dari jalan kegelapan menuju ke jalan terang benerang.

Dalam proses penyusunan skripsi ini, penulis banyak mendapat bimbingan dan arahan dari berbagai pihak. Untuk itu ucapan terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya penulis sampaikan terutama kepada:

1. Prof. Dr. H. Abd. Haris, M.Ag., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Usman Pagalay, M.Si., selaku Ketua Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Ari Kusumastuti, M.Si., M.Pd., selaku dosen pembimbing I yang telah banyak memberikan arahan, nasihat, dan berbagi pengalaman berharga kepada penulis.
5. Angga Dwi Mulyanto, M.Si., selaku dosen pembimbing II yang telah banyak memberikan arahan, nasihat, dan berbagi pengalaman berharga kepada penulis.

6. Segenap civitas akademika Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang terutama seluruh dosen, terima kasih atas segala ilmu dan bimbingannya.
7. Ayah dan Ibu yang selalu memberikan doa, semangat, serta motivasi kepada penulis sampai saat ini.
8. Teman-teman jurusan Matematika angkatan 2016 yang bersama-sama untuk saling mendukung dan berjuang mencapai mimpi dan cita-cita.
9. Sahabat-sahabat penulis terutama “Qintifly Brotherhood” dan “Seduler Kost Bu Siti” yang telah banyak menerima keluh kesah dan ghibah selama jauh dari orang tua serta memberi dukungan moral untuk berjuang bersama.
10. Semua pihak yang ikut serta membantu dalam menyelesaikan skripsi ini baik materil maupun moril.

Akhirnya penulis berharap semoga skripsi ini bermanfaat bagi penulis dan bagi pembaca.

Wassalamu'alaikum Wr. Wb.

Malang, 17 Juni 2020

Penulis

DAFTAR ISI

| | |
|---|-------------|
| HALAHAMAN JUDUL | |
| HALAMAN PENGAJUAN | |
| HALAMAN PERSETUJUAN | |
| HALAMAN PENGESAHAN | |
| HALAMAN PERNYATAAN KEASLIAN TULISAN | |
| HALAMAN MOTO | |
| HALAMAN PERSEMBAHAN | |
| KATA PENGANTAR..... | viii |
| DAFTAR ISI..... | x |
| DAFTAR TABEL | xii |
| DAFTAR GAMBAR..... | xiii |
| ABSTRAK | xiv |
| ABSTRACT | xv |
| ملخص | xvi |
| | |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 5 |
| 1.3 Tujuan Penelitian..... | 5 |
| 1.4 Manfaat Penelitian..... | 6 |
| 1.5 Batasan Masalah..... | 6 |
| 1.6 Sistematika Penulisan..... | 6 |
| | |
| BAB II TINJAUAN PUSTAKA..... | 8 |
| 2.1 Pembahasan Tentang Sistem Pendukung Keputusan..... | 8 |
| 2.1.1 Konsep Islam Tentang Keadilan Dalam Mengambil Keputusan...8 | 8 |
| 2.1.2 Sistem Pendukung Keputusan..... | 9 |
| 2.1.3 Karakteristik Sistem Pendukung Keputusan..... | 10 |
| 2.2 Metode TOPSIS..... | 12 |
| 2.2.1 Unsur-Unsur Metode TOPSIS..... | 12 |
| 2.2.2 Tahapan Perhitungan Metode TOPSIS..... | 14 |
| 2.3 Python..... | 16 |
| 2.4 <i>Game</i> | 19 |
| 2.5 <i>Data Filtering</i> | 21 |

| | |
|--|-----------|
| BAB III METODE PENELITIAN | 22 |
| 3.1 Jenis dan Sumber Data..... | 22 |
| 3.2 Variabel Penelitian..... | 22 |
| 3.3 Metode Analisis Data | 22 |
| 3.3.1 Persiapan Penelitian..... | 22 |
| 3.3.2 Analisis Data | 23 |
| 3.4 Flowchart Analisis Data | 24 |
| BAB IV HASIL DAN PEMBAHASAN | 26 |
| 4.1 Hasil..... | 26 |
| 4.1.1 <i>Data Filtering</i> | 26 |
| 4.1.2 Perhitungan Manual Metode TOPSIS | 27 |
| 4.1.3 Pemrograman Perhitungan Metode TOPSIS pada Python..... | 38 |
| 4.1.4 Aplikasi Metode TOPSIS pada Python | 46 |
| 4.2 Pembahasan | 49 |
| BAB V PENUTUP..... | 51 |
| 5.1 Kesimpulan | 51 |
| 5.2 Saran | 52 |
| DAFTAR RUJUKAN | 54 |
| LAMPIRAN-LAMPIRAN | |
| RIWAYAT HIDUP | |
| BUKTI KONSULTASI SKRIPSI | |

DAFTAR TABEL

| | | |
|-------------|---|----|
| Tabel 4. 1 | Data Asli | 26 |
| Tabel 4. 2 | Data Setelah Disaring..... | 27 |
| Tabel 4. 3 | Data | 28 |
| Tabel 4. 4 | Bobot Kepentingan Kriteria..... | 29 |
| Tabel 4. 5 | Memberi Kriteria <i>Benefit</i> Atau <i>Cost</i> Terhadap Kriteria Atribut..... | 29 |
| Tabel 4. 6 | Lanjutan Memberi Kriteria <i>Benefit</i> Atau <i>Cost</i> Terhadap Kriteria Atribut | 30 |
| Tabel 4. 7 | Matriks Pembagi | 31 |
| Tabel 4. 8 | Matriks Ternormalisasi | 32 |
| Tabel 4. 9 | Matriks Ternormalisasi Terbobot | 33 |
| Tabel 4. 10 | Lanjutan Matriks Ternormalisasi Terbobot | 34 |
| Tabel 4. 11 | Matriks A+ dan A- | 35 |
| Tabel 4. 12 | Hasil Perhitungan D+ dan D-..... | 37 |
| Tabel 4. 13 | Preferensi Untuk Setiap Kriteria | 38 |
| Tabel 4. 14 | Hasil Perhitungan Manual Metode TOPSIS | 49 |

DAFTAR GAMBAR

| | | |
|--------------|--|----|
| Gambar 3. 1 | <i>Flowchart</i> Analisis Data | 24 |
| Gambar 3. 2 | Lanjutan <i>Flowchart</i> Analisis Data | 25 |
| Gambar 4. 1 | <i>Output</i> Matriks Pembagi pada <i>Jupyter Notebook</i> | 40 |
| Gambar 4. 2 | <i>Output</i> Matriks Ternormalisasi pada <i>Jupyter Notebook</i> | 41 |
| Gambar 4. 3 | <i>Output</i> Matriks Ternormalisasi Terbobot pada <i>Jupyter Notebook</i> | 42 |
| Gambar 4. 4 | <i>Output</i> Solusi Ideal Positif A+ pada <i>Jupyter Notebook</i> | 43 |
| Gambar 4. 5 | <i>Output</i> Solusi Ideal Negatif A- pada <i>Jupyter Notebook</i> | 43 |
| Gambar 4. 6 | <i>Output</i> D+ pada <i>Jupyter Notebook</i> | 44 |
| Gambar 4. 7 | <i>Output</i> D- pada <i>Jupyter Notebook</i> | 45 |
| Gambar 4. 8 | <i>Output</i> Hasil Preferensi pada <i>Jupyter Notebook</i> | 45 |
| Gambar 4. 9 | <i>Output</i> Hasil Perangkingan Alternatif pada <i>Jupyter Notebook</i> | 46 |
| Gambar 4. 10 | Tahap Input Variabel Data pada Aplikasi | 47 |
| Gambar 4. 11 | Tahap Input Nilai Data..... | 48 |
| Gambar 4. 12 | Hasil Akhir dan Perangkingan | 48 |

ABSTRAK

Pahlevy, Mohammad Rifqi Riza. 2020. **Aplikasi Metode TOPSIS Pada Python Untuk Menentukan *Most Valuable Player* Turnamen *Mobile Legends Professional League Indonesia Season 5***. Skripsi. Jurusan Matematika Fakultas Sains Dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ari Kusumastuti, M. Si., M. Pd. (II) Angga Dwi Mulyanto, M.Si.

Kata Kunci: Atribut, *Game*, Kriteria, MVP, TOPSIS.

Melesatnya perkembangan teknologi sudah tidak bisa dibendung lagi. Salah satunya yaitu *game*. *Game* sudah diakui oleh pemerintah sebagai salah satu cabang olahraga yang biasa dikenal dengan *e-sport*. Salah satu *game* yang sedang naik daun yaitu *Mobile Legends: Bang-Bang* (MLBB). *Mobile Legends Professional League* (MPL) merupakan turnamen MLBB di kancah professional yang memasuk gelaran musim ke 5. Diadakan pada tanggal 8 Februari 2020 sampai tanggal 12 April 2020 dan setelah selesai turnamen akan dipilih *Most Valuable Player* (MVP) atau pemain terbaik dari turnamen MPL. Pemilihan MVP turnamen MPL menggunakan metode TOPSIS. Metode TOPSIS merupakan salah satu metode SPK (sistem pembantu keputusan). Metode TOPSIS adalah metode pengambilan keputusan multikriteria yang memiliki konsep dimana memilih alternatif terbaik yang memiliki jarak terpendek dari solusi ideal positif dan jarak terpanjang dari solusi ideal negatif. Bahasa pemrograman python digunakan untuk membuat aplikasi metode TOPSIS. Algoritma pemrograman metode TOPSIS yaitu: input data, pembobotan kriteria, mengasumsikan kriteria apakah termasuk *cost/benefit*, normalisasi data, pembobotan data ternormalisasi, menghitung solusi ideal positif dan negatif, menghitung jarak antara alternatif dengan solusi ideal positif dan negatif serta menghitung nilai akhir dan merangkingnya. Alternatif yang digunakan merupakan 10 pemain dari kedua tim finalis mpl yaitu: BAJAN, LUMINAIRE, REKT, REXXY, WANN, LEMON, LJ, R7, VYN dan XIN. Ketujuh kriteria yang digunakan dalam menentukan MVP dalam game MLBB meliputi: *KPG*, *DPG*, *APG*, *GPM*, *KP%*, *DPM* dan *TANKED*. Hasil perhitungan metode TOPSIS yang didapat yaitu XIN sebagai MVP MPL ID *season 5* dengan hasil akhir 0,6398.

ABSTRACT

Pahlevy, Mohammad Rifqi Riza. 2020. **TOPSIS METHOD IN DETERMINING THE MOST VALUABLE PLAYER TOURNAMENT MPL ID SEASON 5 AND APPLICATION IN PYTHON.** Thesis. Department Of Mathematics, Faculty Of Science And Technology, Maulana Malik Ibrahim State Islamic University Of Malang. Advisors: (I) Ari Kusumastuti, M. Si., M. Pd. (II) Angga Dwi Mulyanto, M. Si.

Keywords: TOPSIS, MVP, Python, Attribute, Criteria

The rapid developments of technology can no longer be stopped. For example *game*. The *games* have been recognized by the government as a sport commonly known as *e-sport*. One game that that rising is *Mobile Legends: Bang-Bang* (MLBB). *Mobile Legends Proffesional League* (MPL) is a MLBB tournament in the professional arena which enters the 5th season event. It is held on February 8th 2020 until April 12nd 2020 and after finished the tournament will be selected *Most Valuable Player* (MVP) from the MPL tournament. The selection of MPL tournament MVP uses the TOPSIS method. The TOPSIS method is one of the DSS (decision support system) methods. The TOPSIS method is a multi-criteria decision making method that has the concept of choosing the best alternative that has the shortest distance from a positive ideal solution and the longest distance from a negative ideal solution. Python programming language is used to create applications in the TOPSIS method. The python programming language is used to create the TOPSIS method application. The TOPSIS method programming algorithm is: data input, criteria weighting, assuming whether the criteria include *cost/benefit*, data normalization, weighted normalized data, calculating positive and negative ideal solutions, calculating the distance between alternatives with positive and negative ideal solutions and calculating the final value and ranking them. The alternative used is 10 players from the two MPL finalist teams, namely: BAJAN, LUMINAIRE, REKT, REXXY, WANN, LEMON, LJ, R7, VYN and XIN. The seven criterias that have been used in determining the MVP in the MLBB game include: *DPG, APG, GPM, KP%, DPM* and *TANKED*. The results of the calculation of the TOPSIS method obtained are XIN as the MVP MPL ID *season 5* with a final result of 0.6398.

ملخص

باهليفي، محمد رفقي ريزا. 2020. برامج طريقة توبسيس عند بيتون لإثبات أفضل لاعب (MVP) في بطولة موبايل ليجند (MPL) مستوى الوطنية الخامسة. البحث الجامعي. قسم الرياضيات كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (1) أري كوسومستوتي، الماجستير، (II) أنجا دوي مولياتنو، الماجستير

الكلمات الرئيسية: الصفة، اللعبة، المعيار، أفضل لاعب، توبسيس

كانت سرعة تطور التكنولوجيا لاتقطع. وأحد أجزاءها هو اللعبة. قد اعترفت الحكومة أن اللعبة هي نوع الرياضة حيث سميت بالرياضة الإلكترونية. وإحدى الألعاب المشهورة اليوم هي موبايل ليجند: بانج-بانج (MLBB). بطولة موبايل ليجند هي بطولة موبايل ليجند: بانج-بانج في عالم الاحتراف التي وصلت إلى الدور الخامس. تقوم تلك البطولة الخامسة من 8 فبراير 2020 إلى 12 أبريل 2020 ويختار فيها أفضل لاعب من البطولة. اختيار أفضل لاعب في البطولة بطريقة توبسيس. طريقة توبسيس هي إحدى الطرق من النظام المساعد في الإثبات (SPK). طريقة توبسيس هي طريقة أخذ القرار على عدة الأوصاف التي لها تصميم أينما أن تختار أفضل البديل وأقرب البعد من الحل التام الإيجابي وأبعد البعد من الحل البعيد السلبي. لغة برمجة بيتون مستخدمة في صنع برامج طريقة توبسيس. خوارزمية برمجة توبسيس هي: إدخال البيانات، ترجيح الصفة، توزيع الصفة على تكلفة أو ربح، تطبيع البيانات، ترجيح البيانات المطبوعة، حساب الحل التام الإيجابي والسلبي، حساب البعد بين البديل والحل التام الإيجابي والسلبي وحساب النتيجة الآخرة وتكوينها. البديل المستخدم هو 10 لاعب من الفريقين المتأهلين إلى الجولة النهائية للبطولة وهم: باجان، لومينايري، ريكت، ريكسي، وان، ليمون، إيلج، 7، فين وسين. سبع مجالات التي تستخدم في إثبات أفضل لاعب هي: KPG, DPG, APG, GPM, KP%, DPM, TANKED. نتيجة الحساب بطريقة توبسيس هي تدل إلى أن سين هو أفضل لاعب في بطولة موبايل ليجند مستوى الوطنية للدور الخامس بنتيجة آخرة 0,6398.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Melesatnya perkembangan teknologi sudah tidak bisa dibendung lagi. Perkembangan yang pesat membuat banyak inovasi-inovasi baru di bidang teknologi masa kini seperti halnya komputer/laptop. Pada awal kemunculan komputer digunakan hanya untuk menghitung (*compute*). Berkembangnya *game* yang merupakan salah satu dari inovasi perkembangan teknologi masa kini. *Game* akan menghasilkan efek positif Jika *game* digunakan secara baik dan bijaksana. Contohnya *game* dapat menurunkan tingkat stress pada diri seseorang (Haristu, 2019).

E-sport merupakan permainan video *game* yang kompetitif. *E-sport* memiliki kepanjangan *electronic sport*, yang berarti adalah olahraga elektronik dalam hal ini adalah *video game* (Haristu, 2019). Pemerintah juga sudah mendukung kemajuan di bidang *e-sport* (olahraga elektronik) ini dengan langkah nyata membuat turnamen-turnamen bertepatan Piala Presiden. Salah satu *game* yang dipertandingkan yaitu MLBB (*Mobile Legends: Bang-Bang*).

Selain itu turnamen-turnamen yang diadakan langsung oleh *developer game* juga sudah sering digelar di Indonesia salah satunya yaitu MPL (*Mobile Legends Professional League*). Musim MPL terakhir yaitu musim kelima yang digelar pada tanggal 8 Februari 2020 sampai tanggal 12 April 2020. Di akhir gelaran MPL diberikan beberapa penghargaan atas pencapaian-pencapaian yang didapatkan oleh

para pemain. Salah satu penghargaan yang diberikan yaitu MVP (*Most Valuable Player*) atau pemain terbaik.

MVP merupakan pemain terbaik dalam sitem *game* MLBB yang mana pemain tersebut memiliki statistik permainan terbaik dalam satu *game*. Atribut-atribut yang dihitung dan digunakan untuk menentukan MVP meliputi; *Kill per Game*, *Death per Game*, *Assist per Game*, *Gold per Game*, *Killing Participation*, *Damage per Minute* dan *Damage Taken per Minute*. Data-data statistik tersebut didapatkan dalam setiap *game* yang dimainkan para *player* dan nantinya akan diakumulasikan di akhir kompetisi untuk mempermudah membaca hasil permainan para *player*.

Metode TOPSIS (*Technique for Order Preference Similarity to Ideal Solution*) merupakan metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang 1981 (Efenie & Hozairi, 2016). Tujuan dari metode TOPSIS yaitu untuk menentukan kedua solusi yaitu solusi ideal positif dan solusi ideal negatif. Pada solusi ideal positif, solusi memaksimalkan kriteria manfaat dan meminimalkan kriteria biaya. Kebalikan dari solusi ideal positif, solusi ideal negatif memaksimalkan kriteria biaya dan meminimalkan kriteria manfaat. Kesimpulan/alternatif yang optimal dalam metode TOPSIS sendiri adalah yang paling dekat dengan solusi ideal positif dan paling jauh dari solusi ideal negatif.

Aplikasi untuk menentukan penjurusan untuk tingkat sekolah SMA dibuat oleh Hidayat. Metode TOPSIS diterapkan dalam perhitungannya untuk menentukan keputusan akhir. Alternatif yang digunakan yaitu jurusan bahasa, IPA dan IPS sedangkan kriteria yang digunakan yaitu nilai-nilai dari semua mata pelajaran yang diajarkan. Hasil perbandingan dengan penjurusan sekolah langsung (tanpa metode

topsis) persentase penjurusan menggunakan metode topsis memiliki tingkat persentase kebenaran 53%.

Penelitian yang dilakukan oleh Karmila menggunakan metode TOPSIS ini dengan tujuan merekomendasikan *smartphone* untuk kalangan pemuda. Menggunakan 6 kriteria yaitu; kamera, desain, kapasitas memori, baterai, harga dan bodi dan 5 alternatif *smartphone* yaitu Vivo V3, Oppo A37, Samsung Galaxy J7 core, Asus Zenfone 3 dan Xiaomi Mi A 1. Kesimpulan yang didapatkan yaitu; 1. Siste TOPSIS dapat digunakann untuk merekomendasikan *smartphone* pilihan di kalangan anak muda, dan 2. Hasil perhitungan yang didapat yaitu 2 nilai tertinggi yaitu 1 untuk Asus Zenfone A3 dan 0,8009 untuk Vivo V3.

Penelitian tentang metode TOPSIS juga dilakukan oleh Sunarti. Bertujuan untuk menentukan rumah tempat tinggal yang nyaman. Alternatif yang digunakan yaitu lima perumahan di kota Depok sedangkan kriteria yang digunakan yaitu harga, fasilitas, lokasi, KPR, tipe rumah, pembayaran awal dan fasilitas. Alternatif terbaiknya yaitu perumahan *Sawangan Permai* dengan nilai 0,73.

Pada dasarnya metode TOPSIS dapat dihitung menggunakan rumus topsis dengan bantuan kalkulator. Namun untuk mempercepat perhitungan dan meningkatkan akurasi hasil perhitungan dapat digunakan aplikasi Microsoft Excel. Selanjutnya untuk men-*generalisasi* metode TOPSIS, maka dibutuhkan aplikasi bantuan bahasa pemrograman Python. Bahasa pemrograman Python ini digunakan untuk membuat program yang mampu membantu menyelesaikan permasalahan pada metode TOPSIS secara *general* tidak terpaku hanya dalam satu persoalan.

Dengan memasukkan kriteria dan alternatif sesuai kebutuhan dan memasukkan nilainya masing-masing sesuai kriteria dan alternatifnya.

Bahasa pemrograman Python merupakan bahasa pemrograman yang dapat dikembangkan oleh siapa saja karena bersifat *Open Source* atau dengan kata lain bahasa pemrograman ini gratis, dapat digunakan tanpa lisensi dan dapat dikembangkan semampu yang dapat dilakukan. Python dirancang untuk memberikan kemudahan kepada *programer* baik dari segi efisiensi waktu, maupun kemudahan dalam pengembangan program dan dalam hal kompatibilitas dengan sistem. Python sendiri juga merupakan *multiplatform* yang banyak digunakan pada aplikasi teknologi saat ini dan mendatang. Pertama kali dibuat oleh Guido van Rossum pada awal tahun 1990 di negeri Belanda sebagai pengganti bahasa pemrograman yang disebut ABC (Harismawan et al., 2018).

Q.S. Al-Maidah ayat (8) yang artinya “*Hai orang-orang yang beriman hendaklah kamu jadi orang-orang yang selalu menegakkan (kebenaran) karena Allah, menjadi saksi dengan adil. Dan janganlah sekali-kali kebencianmu terhadap sesuatu kaum, mendorong kamu untuk berlaku tidak adil. Berlaku adillah, karena adil itu lebih dekat kepada takwa. Dan bertakwalah kepada Allah, sesungguhnya Allah Maha Mengetahui apa yang kamu kerjakan*”. Ayat tersebut menjelaskan tentang keadilan yang wajib ditegakkan oleh semua manusia tanpa pandang bulu. Hal ini berkaitan dengan penentuan MVP menggunakan metode TOPSIS. Tidak subjektif memilih suatu *player* dari tim tertentu melainkan menilai semua perolehan atribut selama perlombaan berlangsung.

Berdasarkan latar belakang tersebut, dalam tugas akhir ini penulis akan membuat aplikasi metode TOPSIS menggunakan bahasa pemrograman Python untuk menentukan MVP atau pemain terbaik dari kompetisi *Mobile Legends Professional League Indonesia season 5*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana tahapan pemrograman python untuk menentukan *Most Valuable Player Mobile Legends Professional League Indonesia season 5*?
2. Bagaimana hasil perhitungan pemrograman python untuk menentukan *Most Valuable Player Mobile Legends Professional League Indonesia season 5*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka tujuan pada penelitian ini adalah:

1. Untuk mengetahui tahapan pemrograman python untuk menentukan *Most Valuable Player Mobile Legends Professional League Indonesia season 5*.
2. Untuk mengetahui hasil perhitungan pemrograman python dalam menentukan *Most Valuable Player Mobile Legends Professional League Indonesia season 5*.

1.4 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Penelitian ini dapat menambah wawasan dan pengetahuan tentang metode TOPSIS dan bagaimana tahapan-tahapan dalam pembuatan pemrogramannya pada Python.
2. Penulis dapat menentukan MVP MPL ID *season 5* menggunakan metode TOPSIS dengan bantuan bahasa pemrograman bahasa Python.

1.5 Batasan Masalah

Agar tidak terjadi perluasan masalah dalam penelitian ini, maka diperlukan adanya batasan masalah sebagai berikut :

1. Data yang digunakan yaitu data *player* dari tim Evos Legends dan Tim RRQ yang lolos babak *grandfinal*.
2. Metode yang digunakan yaitu metode TOPSIS (*Technique for Order Preference Similarity to Ideal Solution*).
3. Bahasa pemrograman yang digunakan yaitu Python.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

Bab I Pendahuluan

Pada bab ini akan diuraikan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian dan sistematika penulisan.

Bab II Kajian Pustaka

Pada bab ini akan dijelaskan teori-teori yang mendasari pembahasan diantaranya; *game*, metode TOPSIS dan Python.

Bab III Metode Penelitian

Pada bab ini akan disajikan tentang jenis dan sumber data yang dipakai, variabel-variabel penelitian, persiapan penelitian, dan Flowchart analisa data.

Bab IV Hasil dan Pembahasan

Pada bab ini merupakan bab inti dari skripsi yang menjelaskan tentang langkah-langkah perhitungan menggunakan metode TOPSIS dan algoritma pemrograman pada aplikasi python dan perhitungannya pada aplikasi metode TOPSIS pada python.

Bab V Penutup

Pada bab ini akan disajikan kesimpulan dan saran dari hasil pembahasan.

BAB II

TINJAUAN PUSTAKA

2.1 Pembahasan Tentang Sistem Pendukung Keputusan

2.1.1 Konsep Islam Tentang Keadilan Dalam Mengambil Keputusan

Konsep tentang adil dan keadilan dalam agama islam mendapatkan porsi yang begitu penting. Banyak sekali ayat yang berisi tentang keadilan. Keadilan sendiri dalam bahasa Arab disebut sebagai *Al- 'Adalah* yang berarti keadilan dalam semua cakupan dan sangat ditekankan dalam agama islam. Dua (2) ayat diantaranya yang menerangkan tentang keadilan yaitu Q.S. Al-Maidah ayat 8 dan Q.S An-Nisa ayat 135:

“Hai orang-orang yang beriman hendaklah kamu jadi orang-orang yang selalu menegakkan (kebenaran) karena Allah, menjadi saksi dengan adil. Dan janganlah sekali-kali kebencianmu terhadap sesuatu kaum, mendorong kamu untuk berlaku tidak adil. Berlaku adillah, karena adil itu lebih dekat kepada takwa. Dan bertakwalah kepada Allah, sesungguhnya Allah Maha Mengetahui apa yang kamu kerjakan”.

“Wahai orang-orang yang beriman, jadilah kamu orang yang benar-benar penegak keadilan, menjadi saksi karena Allah biarpun terhadap dirimu sendiri atau ibu bapa dan kaum kerabatmu. Jika ia kaya ataupun miskin, maka Allah lebih tahu kemaslahatannya. Maka janganlah kamu mengikuti hawa nafsu karena ingin menyimpang dari kebenaran. Dan jika kamu memutar balikkan (kata-kata) atau enggan menjadi saksi, maka sesungguhnya Allah adalah Maha Mengetahui segala apa yang kamu kerjakan” (Q.S. An Nisa' 4:135).

Dijelaskan dari kedua ayat tersebut bahwa manusia terutama orang-orang yang beriman memang harus bersikap adil. Karena jika tidak adil keimanannya perlu ditinjau kembali. Selain itu kata *Al'Adl* adalah kata tauhid karena merupakan salah satu nama Allah. Adil berarti jalan yang lurus dan sikapnya selalu menggunakan urutan yang sama. Persamaan itulah yang menjadikan seorang yang menegakkan keadilan haruslah berpihak kepada semua (kebenaran) dan bukan hanya berpihak hanya pada salah satu.

2.1.2 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur dimana tak seorangpun tahu secara pasti keputusan tersebut seharusnya dibuat (Azizi, 2014).

Sistem pendukung keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan peringkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. SPK juga bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik (Sari et al., 2018).

SPK merupakan implementasi teori-teori pengambilan keputusan yang telah diperkenalkan oleh ilmu-ilmu seperti *operation research* dan *management science*. Hanya saja terdapat perbedaan dimana jika dahulu untuk mencari penyelesaian masalah yang dihadapi harus dilakukan perhitungan iterasi secara manual namun untuk saat ini dapat dihitung dengan computer karena kemampuannya untuk menyelesaikan persoalan yang sama dalam waktu yang relatif singkat.

2.1.3 Karakteristik Sistem Pendukung Keputusan

Berdasarkan pengertian Sistem Pendukung keputusan (SPK) maka dapat ditentukan karakteristik dari SPK diantaranya:

1. Perancangan SPK yaitu untuk membantu seseorang dalam mengambil keputusan yang sifat masalahnya semi terstruktur ataupun tidak terstruktur dengan menambahkan kebijaksanaan manusia dan informasi komputerisasi.
2. SPK dalam proses pengolahannya mengkombinasikan penggunaan model-model analisis dengan Teknik pemasukan data konvensional serta fungsi-fungsi pencari informasi.
3. SPK dibuat dan dirancang sedemikian rupa sehingga dapat digunakan dan dioperasikan dengan mudah.
4. Perancangan SPK dengan menekankan aspek fleksibilitas dan kemampuan adaptasi yang tinggi.

Berdasarkan karakteristiknya didapat manfaat dan keuntungan SPK bagi pemakai yaitu: (Azizi, 2014)

1. SPK memperluas kemampuan pengambilan keputusan dalam memproses data atau informasi bagi pemakainya.
2. SPK membantu seorang dalam mengambil keputusan untuk memecahkan berbagai masalah yang sangat tidak terstruktur.
3. SPK dapat menghasilkan solusi dengan lebih cepat dan hasilnya akurat.
4. Pada dasarnya SPK tidak bisa memecahkan masalah yang dihadapi pengambil keputusan namun dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya karena mampu menyajikan berbagai alternatif pemecahan masalah.

Selain memiliki manfaat dan keuntungan, SPK juga memiliki keterbatasan, yaitu: (Azizi, 2014)

1. Beberapa kemampuan manusia seperti manajemen atau bakat tidak dapat dimodelkan sehingga model yang nanti dihasilkan tidak mencerminkan permasalahan sebenarnya.
2. Kemampuan SPK terbatas pada pengetahuan dasar dan modal dasar yang dimilikinya.
3. Proses-proses SPK juga dilakukan bergantung perangkat lunak yang digunakan.
4. SPK tidak mempunyai intuisi seperti manusia. Sehingga sistem ini dirancang hanya untuk membantu seseorang dalam mengambil keputusan.

2.2 Metode TOPSIS

TOPSIS (*Technique For Order Preference By Similarity To Ideal Solution*) adalah salah satu metode pengambilan keputusan multikriteria yang memiliki konsep dimana alternatif yang terpilih merupakan alternatif terbaik yang memiliki jarak terpendek dari solusi ideal positif dan jarak terpanjang dari solusi ideal negatif. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. Hal ini disebabkan konsepnya yang sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan mengukur kinerja relatif dari alternatif – alternatif keputusan dalam bentuk matematis yang sederhana (Karmila et al., 2017).

2.2.1 Unsur-Unsur Metode TOPSIS

Unsur-unsur yang harus ada untuk perhitungan dalam metode TOPSIS yaitu (Alamsyah, 2017):

1. Alternatif

Alternatif merupakan subjek-subjek yang akan dihitung.

2. Kriteria

Kriteria merupakan sifat-sifat yang dimiliki oleh alternatif. Dapat dikatakan kriteria merupakan apa saja yang dihitung dari alternatif.

3. Bobot kriteria

Bobot kriteria merupakan bobot kepentingan dari kriteria-kriteria yang ada. Nilai bobot harus sesuai dengan seberapa penting unsur kriteria.

4. *Cost/Benefit*

Kriteria dalam metode TOPSIS dikatakan *benefit* maka semakin besar nilai semakin bagus dan jika semakin kecil nilainya semakin buruk. Sebaliknya jika *cost* jika nilainya semakin besar maka dinilai semakin rendah dan jika semakin kecil maka nilainya semakin tinggi.

5. Data Ternormalisasi

Normalisasi data merupakan cara transformasi data agar distribusinya menjadi normal. Sehingga data ternormalisasi yaitu data yang tertransformasi dan distribusi datanya sudah normal.

6. A^+ dan A^-

Solusi ideal positif (A^+) merupakan nilai alternatif tertinggi dari suatu kriteria yang sifatnya *benefit* dan nilai alternatif terendah dari suatu kriteria yang sifatnya *cost*. Sebaliknya solusi ideal negatif (A^-) merupakan nilai alternatif terendah dari suatu kriteria yang sifatnya *benefit* dan nilai alternatif tertinggi dari suatu kriteria yang sifatnya *cost*.

7. D^+ dan D^-

Jarak antara alternatif dengan solusi ideal positif (D^+) dan jarak antara alternatif dengan solusi ideal negatif (D^-) dihitung untuk menentukan nilai preferensi atau hasil akhir.

8. Nilai preferensi

Hasil akhir dari perhitungan metode TOPSIS. Setelah didapat nilai preferensi dilakukan perbandingan alternatif dari yang tertinggi ke terendah.

2.2.2 Tahapan Perhitungan Metode TOPSIS

Adapun tahapan – tahapan dalam metode TOPSIS yaitu:

1. membuat data keputusan yang ternormalisasi
2. membuat data keputusan yang ternormalisasi tebobot
3. menentukan data solusi ideal positif dan data solusi ideal negatif
4. menentukan jarak antara nilai setiap alternatif dengan data solusi ideal positif dan data solusi ideal negatif
5. menentukan nilai preferensi untuk setiap alternatif

Langkah-langkah untuk menyelesaikan permasalahan menggunakan metode TOPSIS adalah (Alamsyah, 2017):

1. Membuat matriks keputusan yang ternormalisasi

Rumus matriks ternormalisasi yaitu:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.1)$$

Dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$.

r_{ij} = matriks keputusan ternormalisasi.

x_{ij} = elemen matriks alternatif i kriteria j

i = alternatif ke- i .

j = kriteria ke- j .

2. Membuat matriks keputusan yang ternormalisasi terbobot.

$$Y = \begin{bmatrix} y_{11} & y_{12} & y_{1j} \\ y_{21} & y_{22} & y_{2j} \\ y_{i1} & y_{i2} & y_{ij} \end{bmatrix} \quad \text{untuk } y_{ij} = w_j \cdot r_{ij} \quad (2.2)$$

Keterangan:

w_j = bobot kriteria ke-j.

r_{ij} = elemen data dari data keputusan yang ternormalisasi.

- Menentukan matriks solusi ideal positif dan matriks solusi ideal negatif yang dinotasikan A^+ dan A^- .

$$A^+ = (y_1^+, y_2^+, \dots, y_i^+) \quad (2.3)$$

$$A^- = (y_1^-, y_2^-, \dots, y_i^-) \quad (2.4)$$

Dimana:

$y_j^+ = \max y_{ij}$ jika j adalah atribut *benefit* (keuntungan)

$\min y_{ij}$ jika j adalah atribut *cost* (biaya)

$y_j^- = \min y_{ij}$ jika j adalah atribut *benefit* (keuntungan)

$\max y_{ij}$ jika j adalah atribut *cost* (biaya)

- Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif (D^+) dan matriks solusi ideal negatif (D^-).

$$D_i^+ = \sqrt{\sum_{i=1}^n (y_i^+ - y_{ij})^2}; i = 1, 2, \dots, m \quad (2.5)$$

$$D_i^- = \sqrt{\sum_{i=1}^n (y_{ij} - y_i^-)^2}; i = 1, 2, \dots, m \quad (2.6)$$

Keterangan:

y_i^+ adalah elemen dari matriks solusi ideal positif.

y_i^- adalah elemen dari matriks solusi ideal negatif.

- Menentukan nilai preferensi untuk setiap alternatif. Dalam hal ini dapat dikatakan proses perangkingan alternatif. Nilai preferensi merupakan

kedekatan suatu alternatif terhadap solusi ideal. Maka dari itu, alternatif terbaik adalah salah satu yang berjarak terpendek terhadap solusi ideal positif dan berjarak terjauh dengan solusi ideal negatif.

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}; i = 1, 2, \dots, m \quad (2.7)$$

Dimana: Nilai V_i yang lebih besar menunjukkan prioritas alternatif.

2.3 Python

Bahasa pemrograman Python dibuat oleh Guido van Rossum pada tahun 1990 di Belanda sebagai pengganti bahasa pemrograman yang disebut ABC. Python merupakan perkembangan yang berasal dari kontribusi-kontribusi dari berbagai sumber dari seluruh dunia dalam bidang pemrograman bahasa meski pada akhirnya Guido lah yang mematenkannya. Pada bahasa pemrograman Python, semua orang dapat mengembangkan bahasa pemrogramannya sendiri hal ini karena Python sendiri mempunyai sifat *Open Source* atau dapat dikatakan bahwa bahasa pemrograman Python ini gratis, dapat digunakan tanpa lisensi, dan dapat dikembangkan semampu yang dapat dilakukan.

Penulisan sintaks yang lebih fleksibel dibandingkan dengan bahasa pemrograman lainnya membuat Python menjadi mudah dipelajari. Selain itu, bahasa pemrograman Python ini memiliki efisiensi tinggi untuk struktur data level tinggi, pemrograman berorientasi objek lebih sederhana tetapi efektif, dapat bekerja pada multi platform, dan dapat digabungkan dengan bahasa pemrograman lain untuk menghasilkan aplikasi yang diinginkan.

Bahasa pemrograman Python adalah bahasa pemrograman yang mudah dibaca dan terstruktur, hal ini karena di gunakannya sistem indentasi. Yaitu memisahkan blok-blok program susunan indentasi. Jadi untuk memasukan sub-sub program dalam suatu blok, sub-sub program tersebut diletakkan satu atau lebih spasi dari kolom suatu blok program.

Menurut (Saragih, 2018) Python memiliki beberapa keunggulan dan beberapa kekurangan. Keunggulan dari Python meliputi:

1. Python mudah digunakan

Python merupakan bahasa pemrograman tingkat tinggi, artinya bahasa Python lebih mendekati bahasa manusia daripada bahasa mesin. Python juga memiliki struktur dasar yang mengharuskan pengguna untuk meng-include sistemnya sebelum membuat program. Perintah-perintah yang digunakan di Python menggunakan bahasa Inggris yang digunakan sehari-hari seperti *print*, *input*, *export*, dll.

2. Kompabilitas dan Kemampuan Tinggi

Python memiliki kemampuan yang tinggi dan mampu menciptakan aplikasi yang sederhana hingga aplikasi yang kompleks. Bahasa pemrograman Python mendukung pemrograman berbasis grafis (*GUI Programming*). Python juga memiliki kelebihan berupa pengalokasian memori secara dinamis.

3. *Platform Independent*

Maksudnya yaitu program yang dibuat bisa dijalankan di sistem operasi apa saja selama di sistem operasi tersebut terdapat *platform* Python (*Interpreter Python*).

4. *Open Source*

Bahasa pemrograman Python bersifat gratis dan anda bisa mengembangkannya secara pribadi ataupun bersama teman-teman kalian dengan membentuk tim.

Sedangkan kekurangan dari Python yaitu:

1. Beberapa penugasan terdapat di luar jangkauan kemampuan Python sehingga Python tidak secepat atau efisien sebagai statis.
2. Akibat *interpreter* pada Python, maka Python bukan merupakan perangkat bantu terbaik untuk pengantar komponen kinerja kritis.
3. Untuk beberapa komponen, Python tidak dapat digunakan sebagai dasar bahasa pemrograman implementasi namun masih bisa bekerja sebagai bagian depan *script interface*.

Pada Python ada beberapa perintah dasar untuk menyelesaikan persamaan-persamaan matematika. Perintah-perintah tersebut bersifat sederhana, namun jika dikembangkan bisa menjadi pemrograman sebagai alat bantu hitung matematika yang rumit sekalipun. Seperti *if*, *else*, *elif (else if)*, *range*, *append*, *len*, *print* dan lain-lain.

2.4 Game

Game adalah permainan yang menggunakan media elektronik, merupakan sebuah hiburan berbentuk multimedia yang di buat semenarik mungkin agar pemain bisa mendapatkan sesuatu sehingga adanya kepuasan batin. *Game* memiliki beberapa jenis *genre* (aliran) yang terkenal yaitu *MOBA (Multiplayer Online Battle Arena)*, *FPS (Fint Person Shooter)*, *Fighting (Street Fighter)*, *Real Time Strategy Games*. Setiap *genre* memiliki bentuk permainan dan cara untuk memenangkan permainannya masing-masing. *Genre* dari *game* Mobile Legends: Bang-Bang (MLBB) adalah MOBA. Dimana dua tim yang saling bermusuhan beradu strategi sehingga salah satu tim pemenang yaitu tim yang berhasil menghancurkan *base* musuhnya terlebih dahulu (Haristu, 2019).

MVP (*Most Valuable Player*) merupakan sistem penentuan pemain terbaik yang ada di dalam *game* Mobile Legends: Bang-Bang (MLBB). Beberapa atribut/perolehan yang akan digunakan dalam menghitung skor untuk menentukan MVP yaitu:

1. *Kill per game*

Kill per game yaitu jumlah *kill* (membunuh) yang diperoleh oleh setiap pemain dalam setiap *game* yang dimainkan. Setiap pemain yang mampu membunuh/mengeliminasi pemain lawan akan mendapat 1 (satu) *kill*.

2. *Death per game*

Death per game yaitu jumlah *death* (mati) yang diperoleh oleh setiap pemain dalam setiap *game* yang dimainkan. Setiap pemain yang dibunuh oleh lawan, maka pemain tersebut terhitung *death* 1 (satu).

3. *Assist per game*

Assist per game yaitu jumlah *assist* (bantuan) yang diperoleh oleh setiap pemain dalam setiap game yang dimainkan. *Assists* didapat dengan cara membantu tim (berpartisipasi) dalam *kill* yang didapat oleh teman satu tim.

4. *Gold per minute*

Gold per minute yaitu jumlah *gold* (satuan uang/*net worth*) akhir yang didapatkan setelah game berakhir lalu dibagi dengan durasi permainan dalam satuan menit (60 detik). *Gold* bisa didapatkan dengan cara *farming* yaitu membunuh satu *minion*, monster *jungle* ataupun dengan menghancurkan *tower* musuh. Selain itu *gold* juga bisa didapat dengan (*kill*) musuh ataupun dengan *assist*.

5. *Killing participation %*

Killing participation percentage (persentase partisipasi membunuh) yaitu seberapa banyak *player* berpartisipasi dalam mengeliminasi (*kill*) lawan. Diperoleh dengan menjumlahkan *assist* dan *kill* yang diperoleh dan membaginya dengan jumlah total *kill* tim.

6. *Damage per minute*

Damage per minute yaitu jumlah *damage* (serangan) yang diberikan kepada musuh. Total *damage* yang diberikan selama pertandingan akan ditotal lalu dibagi dengan durasi permainan dalam satuan menit (60 detik).

7. *Damage Taken per minute*

Damage taken per minute yaitu jumlah *damage taken* (serangan diterima) yang diterima dari musuh. Total *damage taken* yang diberikan selama pertandingan akan ditotal lalu dibagi dengan durasi permainan dalam satuan menit (60 detik).

2.5 Data Filtering

Filter dalam bahasa Indonesia berarti saringan dan kata kerjanya yaitu menyaring. *Filtering data* yaitu kegiatan menyaring data yang dalam artian aplikasi yaitu menampilkan baris data tertentu sesuai *filter* yang diterapkan dan menyembunyikan sisa baris data lainnya yang tidak digunakan. Proses *data filtering* berguna untuk mengurangi jumlah data yang ditampilkan dan membuat data menjadi lebih ringkas sehingga mempermudah dalam proses perhitungan.

BAB III

METODE PENELITIAN

3.1 Jenis dan Sumber Data

Penelitian ini menggunakan data sekunder yang didapatkan dari www.id.mpl.com. Data sekunder yang digunakan pada penelitian ini adalah data statistik atribut *player* dari kedua tim *grandfinalis* MPL ID *season* 5 yaitu dari Team Evos Legends dan Team RRQ yang mana berlangsung pada hari Minggu tanggal 12 April 2020.

3.2 Variabel Penelitian

Variabel bebas yang digunakan yaitu 10 pemain yang akan bertanding di babak *grandfinal* yaitu: Bajan, Luminaire, Rexxy, REKT, Wann, Xin, Lemon, LJ, R7, Vyn. Sedangkan variabel terikat yang digunakan dalam penelitian ini yaitu; *Kill Per Game* (KPG), *Death Per Game* (DPG), *Assist Per Game* (APG), *Gold Per Minute* (GPM), *Killing Participation Percentage* (KP%), *Damage Per Minute* (DPM) dan *Damage Taken Per Minute* (TANKED).

3.3 Metode Analisis Data

3.3.1 Persiapan Penelitian

Dalam penelitian ini, peneliti melakukan pendekatan kajian literatur dan simulasi komputasi. Literatur digunakan untuk memahami teori dan konsep dari

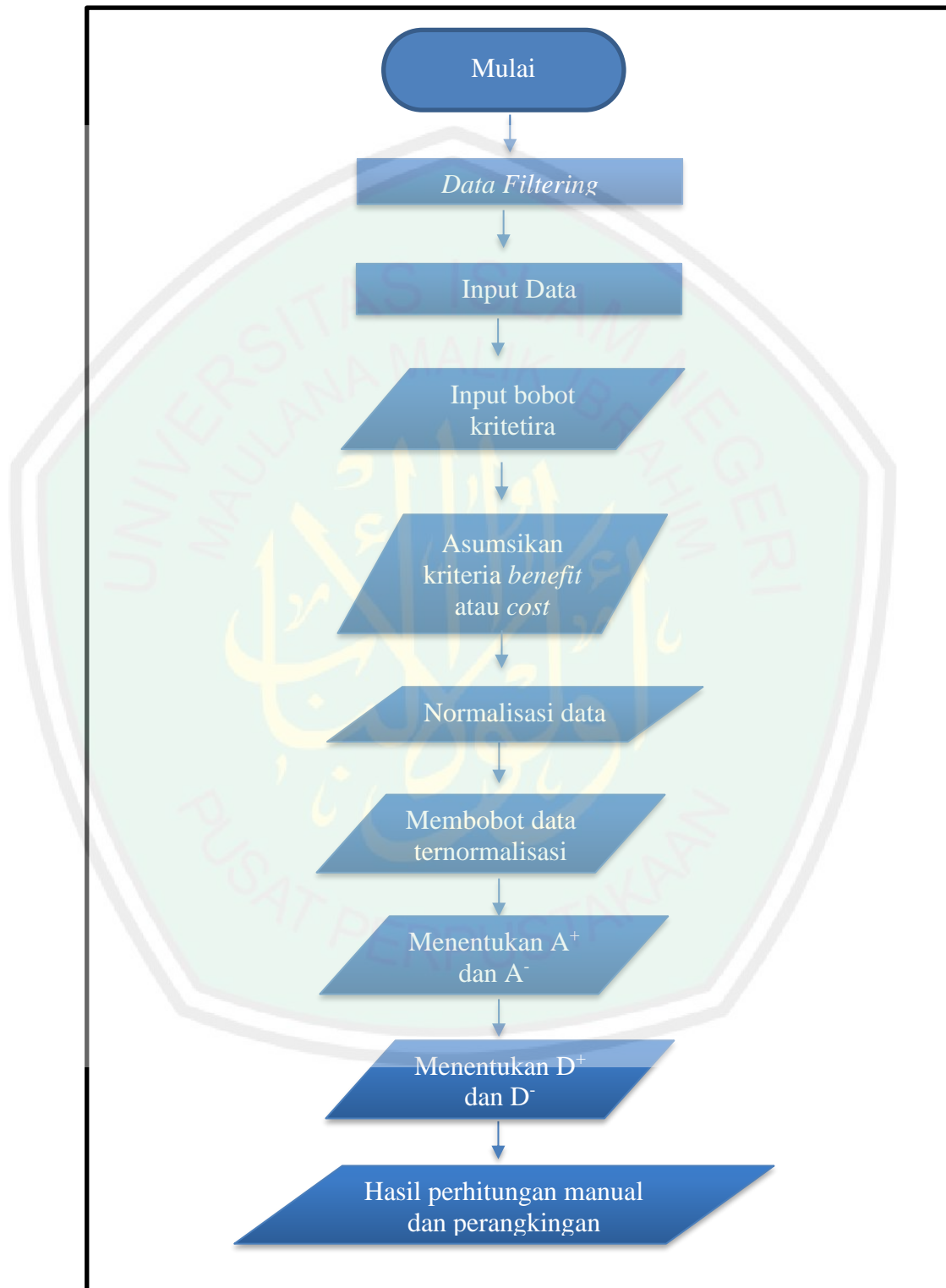
Metode TOPSIS. Dari hasil perhitungan secara manual nantinya akan program metode TOPSIS akan dibuat dengan bantuan bahasa pemrograman Python.

3.3.2 Analisis Data

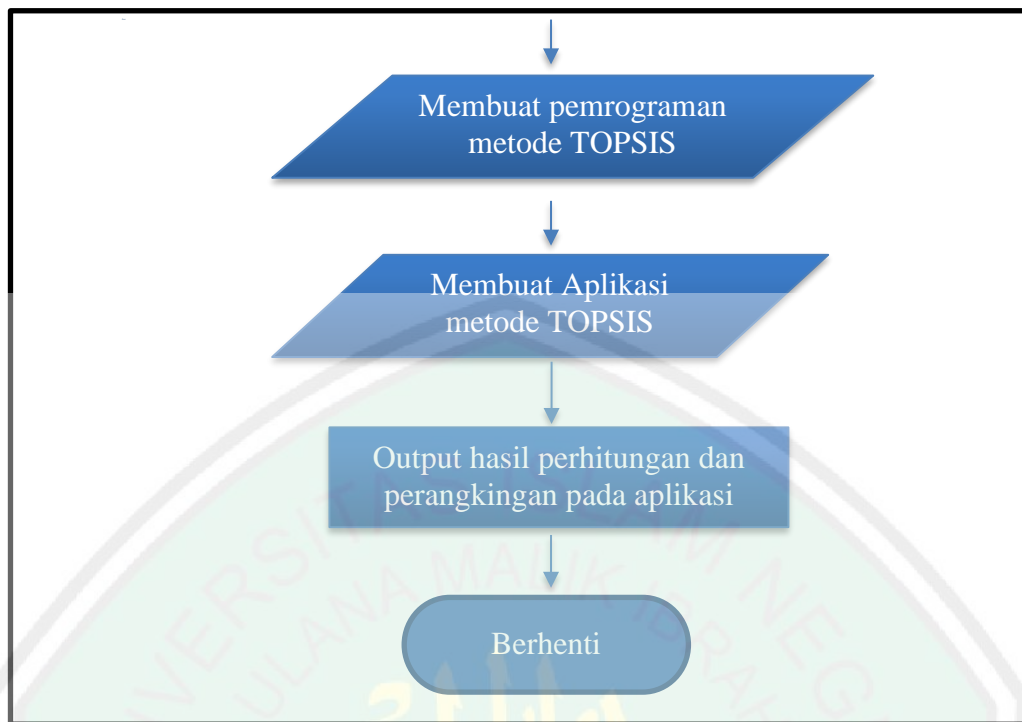
Adapun langkah-langkah yang digunakan pada penelitian ini, yaitu:

1. *Data filtering*.
2. Input data.
3. Menentukan bobot kriteria.
4. Mengasumsikan semua kriteria apakah termasuk kriteria keuntungan (*benefit*) atau termasuk kriteria biaya (*cost*).
5. Menentukan matriks keputusan ternormalisasi.
6. Membuat matriks keputusan ternormalisasi terbobot.
7. Menentukan solusi ideal positif A^+ dan solusi ideal negatif A^- .
8. Menentukan jarak antara nilai terbobot setiap alternatif terhadap solusi ideal positif D^+ dan solusi ideal negatif D^- .
9. Menentukan preferensi (hasil) setiap alternatif.
10. Perangkingan alternatif.

3.4 Flowchart Analisis Data



Gambar 3. 1 *Flowchart* Analisis Data



Gambar 3. 2 Lanjutan *Flowchart* Analisis Data

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Data Filtering

Data filtering dilakukan untuk meringkas data. Data yang ditampilkan tidak hanya meliputi data yang diperlukan untuk menentukan MVP melainkan statistik *player* sepenuhnya untuk memberikan detail hasil turnamen secara menyeluruh selama babak *grandfinal*. Berikut data asli statistik para pemain.

Tabel 4. 1 Data Asli

| <i>PLAYER</i> | <i>GAMES</i> | <i>KPG</i> | <i>DPG</i> | <i>APG</i> | ... | <i>KILLS</i> | <i>DEATHS</i> | <i>ASSISTS</i> |
|---------------|--------------|------------|------------|------------|-----|--------------|---------------|----------------|
| BAJAN | 15 | 1.3 | 2.5 | 6.7 | ... | 19 | 37 | 101 |
| LUMINAIRE | 15 | 1.5 | 2.6 | 7.7 | ... | 22 | 39 | 116 |
| REKT | 15 | 1.7 | 2.4 | 3.6 | ... | 26 | 36 | 54 |
| REXXY | 15 | 2.9 | 1.8 | 5.9 | ... | 44 | 27 | 88 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| WARLORD | 2 | 0.5 | 3.5 | 3.5 | ... | 1 | 7 | 7 |
| PSYCHOO | 2 | 1.5 | 3.5 | 2 | ... | 3 | 7 | 4 |
| WATT | 2 | 6 | 2 | 5.5 | ... | 12 | 4 | 11 |
| LORDGOV | 1 | 0 | 4 | 2 | ... | 0 | 4 | 2 |

Statistik data *player* yang awalnya memiliki 12 kriteria diringkas menjadi 7 kriteria. Begitupun juga pada jumlah *player* yang mana dalam hal ini dikatakan sebagai alternatif yang awalnya sebanyak 33 *player* diringkas menjadi 10 *player*. Untuk melihat data secara keseluruhan dapat dilihat pada lampiran 1. Proses penyaringan ini didasarkan pada jumlah *games* yang dimainkan dan juga *player*

yang tim nya berhasil melaju ke babak *grandfinal*. Sehingga didapat tabel data hasil *filter* (tersaring) dibawah.

Tabel 4. 2 Data Setelah Disaring

| Alternatif (player) | Kriteria | | | | | | |
|------------------------|----------|-----|-----|-----|-----|------|--------|
| | KPG | DPG | APG | GPM | KP% | DPM | TANKED |
| BAJAN | 1,3 | 2,5 | 6,7 | 545 | 62 | 1413 | 6461 |
| LUMINAIRE | 1,5 | 2,6 | 7,7 | 545 | 75 | 2509 | 1975 |
| REKT | 1,7 | 2,4 | 3,6 | 668 | 40 | 1906 | 4770 |
| REXXY | 2,9 | 1,8 | 5,9 | 578 | 69 | 3360 | 1818 |
| WANN | 5,3 | 2,3 | 3,7 | 849 | 69 | 3627 | 2971 |
| LEMON | 2,4 | 2,3 | 4,3 | 690 | 52 | 1901 | 4324 |
| LJ | 0,8 | 2,9 | 6 | 564 | 54 | 1138 | 6502 |
| R7 | 3,4 | 1,3 | 2,8 | 706 | 45 | 1636 | 3427 |
| VYN | 1,7 | 2,5 | 5,8 | 563 | 54 | 2353 | 2474 |
| XIN | 4,9 | 1,7 | 4,5 | 892 | 70 | 4820 | 2408 |

4.1.2 Perhitungan Manual Metode TOPSIS

Langkah-langkah perhitungan TOPSIS yaitu:

1. Menginput Data

Langkah pertama yaitu menginput data. Nilai dari data dimasukkan sesuai dengan kriteria dan alternatif masing-masing.

Tabel 4. 3 Data

| Alternatif (player) | Kriteria | | | | | | |
|------------------------|----------|-----|-----|-----|-----|------|--------|
| | KPG | DPG | APG | GPM | KP% | DPM | TANKED |
| BAJAN | 1,3 | 2,5 | 6,7 | 545 | 62 | 1413 | 6461 |
| LUMINAIRE | 1,5 | 2,6 | 7,7 | 545 | 75 | 2509 | 1975 |
| REKT | 1,7 | 2,4 | 3,6 | 668 | 40 | 1906 | 4770 |
| REXXY | 2,9 | 1,8 | 5,9 | 578 | 69 | 3360 | 1818 |
| WANN | 5,3 | 2,3 | 3,7 | 849 | 69 | 3627 | 2971 |
| LEMON | 2,4 | 2,3 | 4,3 | 690 | 52 | 1901 | 4324 |
| LJ | 0,8 | 2,9 | 6 | 564 | 54 | 1138 | 6502 |
| R7 | 3,4 | 1,3 | 2,8 | 706 | 45 | 1636 | 3427 |
| VYN | 1,7 | 2,5 | 5,8 | 563 | 54 | 2353 | 2474 |
| XIN | 4,9 | 1,7 | 4,5 | 892 | 70 | 4820 | 2408 |

2. Menentukan Bobot Kepentingan Kriteria

Setiap kriteria atribut *player* memiliki bobot kepentingan masing-masing sesuai kriteria yang jika dijumlahkan hasilnya yaitu 100%. Semakin besar bobot

kriteria artinya semakin penting kriteria tersebut dalam menentukan MVP dan juga sebaliknya.

Tabel 4. 4 Bobot Kepentingan Kriteria

| Kriteria | Bobot Kriteria |
|-------------------------|----------------|
| Kill per Game | 10% |
| Death per Game | 10% |
| Assist per Game | 10% |
| Gold per Minute | 20% |
| Kills Participation | 20% |
| Damage Taken per Minute | 15% |
| Tanked | 15% |
| TOTAL | 100% |

3. Mengasumsikan *Benefit* atau *Cost* dari kriteria atribut *player*

Setiap kriteria harus diasumsikan *cost* atau *benefit*. Pengasumsian ini nantinya digunakan untuk membantu menghitung jarak ideal positif maupun jarak ideal negatif. Dikatakan termasuk kriteria *benefit* jika semakin tinggi nilainya semakin bagus juga sebaliknya dan dikatakan termasuk kriteria *cost* jika semakin besar nilainya maka semakin buruk juga sebaliknya.

Tabel 4. 5 Memberi Kriteria *Benefit* atau *Cost* Terhadap Kriteria Atribut

| Alternatif (player) | Kriteria | | | | | | |
|------------------------|----------|------|---------|---------|---------|---------|---------|
| | benefit | cost | benefit | benefit | benefit | benefit | benefit |
| | KPG | DPG | APG | GPM | KP% | DPM | TANKED |
| BAJAN | 1,3 | 2,5 | 6,7 | 545 | 62 | 1413 | 6461 |

Tabel 4. 6 Lanjutan Memberi Kriteria *Benefit* atau *Cost* Terhadap Kriteria Atribut

| | | | | | | | |
|------------|-----|-----|-----|-----|----|------|------|
| LIUMINAIRE | 1,5 | 2,6 | 7,7 | 545 | 75 | 2509 | 1975 |
| REKT | 1,7 | 2,4 | 3,6 | 668 | 40 | 1906 | 4770 |
| REXXY | 2,9 | 1,8 | 5,9 | 578 | 69 | 3360 | 1818 |
| WANN | 5,3 | 2,3 | 3,7 | 849 | 69 | 3627 | 2971 |
| LEMON | 2,4 | 2,3 | 4,3 | 690 | 52 | 1901 | 4324 |
| LJ | 0,8 | 2,9 | 6 | 564 | 54 | 1138 | 6502 |
| R7 | 3,4 | 1,3 | 2,8 | 706 | 45 | 1636 | 3427 |
| VYN | 1,7 | 2,5 | 5,8 | 563 | 54 | 2353 | 2474 |
| XIN | 4,9 | 1,7 | 4,5 | 892 | 70 | 4820 | 2408 |

4. Menentukan Matriks Ternormalisasi

Untuk membantu dalam perhitungan normalisasi data, terlebih dahulu dihitung matriks pembagi menggunakan persamaan (2.1) $\sqrt{\sum_{i=1}^m x_{ij}^2}$ diambil bagian penyebut (pembagi) saja .

$$p_j = \sqrt{\sum_{i=1}^m x_{ij}^2}, p_1 = \sqrt{1,3^2 + 1,5^2 + \dots + 4,9^2} = 9,3910$$

$$p_2 = \sqrt{2,5^2 + 2,6^2 + \dots + 1,7^2} = 7,1993$$

⋮

$$p_7 = \sqrt{6461^2 + 1975^2 + \dots + 2408^2} = 12851,71$$

Sehingga didapatkan matriks pembagi berikut:

Tabel 4. 7 Matriks Pembagi

| | | | | | | |
|--------|--------|---------|----------|----------|-----------|------------|
| 9,3910 | 7,1993 | 16,7887 | 2121,420 | 189,8736 | 8527,8110 | 12851,7100 |
|--------|--------|---------|----------|----------|-----------|------------|

Setelah didapat matriks pembaginya, dihitung matriks ternormalisasi

menggunakan persamaan (2.1) $r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} = \frac{x_{ij}}{p_j}$.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} = \frac{x_{ij}}{p_j}, \quad r_{11} = \frac{x_{11}}{p_1} = \frac{1,3}{9,3910} = 0,1384$$

$$r_{21} = \frac{x_{21}}{p_1} = \frac{1,5}{9,3910} = 0,1957$$

⋮

$$r_{12} = \frac{x_{12}}{p_2} = \frac{2,5}{7,1993} = 0,3473$$

$$r_{22} = \frac{x_{22}}{p_2} = \frac{2,6}{7,1993} = 0,3611$$

⋮

$$r_{107} = \frac{x_{107}}{p_7} = \frac{2408}{12851,71} = 0,1874$$

Sehingga matriks ternormalisasi yang didapat yaitu:

Tabel 4. 8 Matriks Ternormalisasi

| Alternatif (player) | Kriteria | | | | | | |
|------------------------|----------|--------|---------|---------|---------|---------|---------|
| | benefit | cost | benefit | benefit | benefit | benefit | benefit |
| | KPG | DPG | APG | GPM | KP% | DPM | TANKED |
| BAJAN | 0,1384 | 0,3473 | 0,3991 | 0,2569 | 0,3265 | 0,1657 | 0,5027 |
| LIUMINAIRE | 0,1597 | 0,3611 | 0,4586 | 0,2569 | 0,3950 | 0,2942 | 0,1537 |
| R E K T | 0,1810 | 0,3334 | 0,2144 | 0,3149 | 0,2107 | 0,2235 | 0,3712 |
| REXXY | 0,3088 | 0,2500 | 0,3514 | 0,2725 | 0,3634 | 0,3940 | 0,1415 |
| WANN | 0,5644 | 0,3195 | 0,2204 | 0,4002 | 0,3634 | 0,4253 | 0,2312 |
| LEMON | 0,2556 | 0,3195 | 0,2561 | 0,3253 | 0,2739 | 0,2229 | 0,3365 |
| LJ | 0,0852 | 0,4028 | 0,3574 | 0,2659 | 0,2844 | 0,1334 | 0,5059 |
| R7 | 0,3621 | 0,1806 | 0,1668 | 0,3328 | 0,2370 | 0,1918 | 0,2667 |
| VYN | 0,1810 | 0,3473 | 0,3455 | 0,2654 | 0,2844 | 0,2759 | 0,1925 |
| XIN | 0,5218 | 0,2361 | 0,2680 | 0,4205 | 0,3687 | 0,5652 | 0,1874 |

5. Menentukan Matriks Ternormalisasi Terbobot

Matriks ternormalisasi terbobot adalah matriks ternormalisasi yang diberikan bobot sesuai dengan bobot kepentingan kriteria. Pemberian bobot menggunakan persamaan (2.2)

$$y_{ij} = w_j \cdot r_{ij}, \quad y_{11} = w_1 \cdot r_{11} = 0,138 \cdot 0,1 = 0,0138$$

$$y_{21} = w_1 \cdot r_{21} = 0,160 \cdot 0,1 = 0,016$$

⋮

$$y_{12} = w_2 \cdot r_{12} = 0,3473 \cdot 0,1 = 0,0347$$

$$y_{22} = w_2 \cdot r_{22} = 0,3611 \cdot 0,1 = 0,0361$$

⋮

$$y_{107} = w_7 \cdot r_{107} = 0,1874 \cdot 0,15 = 0,0281$$

Sehingga didapat matriks ternormalisasi terbobot berikut:

Tabel 4. 9 Matriks Ternormalisasi Terbobot

| Alternatif (player) | Kriteria | | | | | | |
|------------------------|----------|--------|---------|---------|---------|---------|---------|
| | benefit | cost | benefit | benefit | benefit | benefit | benefit |
| | KPG | DPG | APG | GPM | KP% | DPM | TANKED |
| BAJAN | 0,0138 | 0,0347 | 0,0399 | 0,0514 | 0,0653 | 0,0249 | 0,0754 |
| LIUMINAIRE | 0,0160 | 0,0361 | 0,0459 | 0,0514 | 0,0790 | 0,0441 | 0,0231 |
| REKT | 0,0181 | 0,0333 | 0,0214 | 0,0630 | 0,0421 | 0,0335 | 0,0557 |
| REXXY | 0,0309 | 0,0250 | 0,0351 | 0,0545 | 0,0727 | 0,0591 | 0,0212 |
| WANN | 0,0564 | 0,0319 | 0,0220 | 0,0800 | 0,0727 | 0,0638 | 0,0347 |
| LEMON | 0,0256 | 0,0319 | 0,0256 | 0,0651 | 0,0548 | 0,0334 | 0,0505 |

Tabel 4. 10 Lanjutan Matriks Ternormalisasi Terbobot

| | | | | | | | |
|-----|--------|--------|--------|--------|--------|--------|--------|
| LJ | 0,0085 | 0,0403 | 0,0357 | 0,0532 | 0,0569 | 0,0200 | 0,0759 |
| R7 | 0,0362 | 0,0181 | 0,0167 | 0,0666 | 0,0474 | 0,0288 | 0,0400 |
| VYN | 0,0181 | 0,0347 | 0,0345 | 0,0531 | 0,0569 | 0,0414 | 0,0289 |
| XIN | 0,0522 | 0,0236 | 0,0268 | 0,0841 | 0,0737 | 0,0848 | 0,0281 |

6. Menentukan Solusi Ideal Positif dan Solusi Ideal Negatif

Berdasarkan persamaan (2.3), pada A^+ elemen y_{ij} akan bernilai maksimum jika j merupakan atribut *benefit* (keuntungan) dan bernilai minimum jika j merupakan atribut *cost* (biaya). Sebaliknya menurut persamaan (2.4), pada A^- elemen y_{ij} akan bernilai maksimum jika j merupakan atribut *cost* (biaya) dan bernilai minimum jika j merupakan atribut *benefit* (keuntungan).

Persamaan 2.3 $A^+ = (y_1^+, y_2^+, \dots, y_i^+)$ untuk mencari solusi ideal positif.

$$y_1^+ = \max (0,0138, 0,0160, 0,0181, 0,0309, \dots, 0,0522) = 0,0564$$

$$y_2^+ = \min (0,0347, 0,0361, 0,0333, 0,0250, \dots, 0,0236) = 0,0181$$

⋮

$$y_7^+ = \max (0,0754, 0,0231, 0,0557, 0,0212, \dots, 0,0281) = 0,0759$$

Persamaan 2.4 $A^- = (y_1^-, y_2^-, \dots, y_i^-)$ untuk mencari solusi ideal negatif.

$$y_1^- = \min (0,0138, 0,0160, 0,0181, 0,0309, \dots, 0,0522) = 0,0085$$

$$y_2^- = \max (0,0347, 0,0361, 0,0333, 0,0250, \dots, 0,0236) = 0,0401$$

⋮

$$y_7^- = \min (0,0754, 0,0231, 0,0557, 0,0212, \dots, 0,0281) = 0,0212$$

Sehingga didapatkan matrik solusi ideal positif A^+ dan solusi ideal negatif A^- sebagai berikut:

Tabel 4. 11 Matriks A^+ dan A^-

| | | | | | | | |
|----|--------|--------|--------|--------|--------|--------|--------|
| A+ | 0,0564 | 0,0181 | 0,0459 | 0,0841 | 0,0790 | 0,0848 | 0,0759 |
| A- | 0,0085 | 0,0403 | 0,0167 | 0,0514 | 0,0421 | 0,0200 | 0,0212 |

7. Menentukan Jarak Antara Nilai Terbobot Setiap Alternatif Terhadap Solusi Ideal Positif dan Solusi Ideal Negatif

Persamaan (2.5) dan persamaan (2.6) digunakan untuk menghitung jarak antara nilai terbobot setiap alternatif terhadap solusi ideal positif D^+ dan solusi ideal negatif D^- .

$$\text{Persamaan (2.5)} \quad D_i^+ = \sqrt{\sum_{i=1}^n (y_i^+ - y_{ij})^2}$$

$$D_1^+ = \sqrt{(y_1^+ - y_{1j})^2}$$

$$= \sqrt{(0,0564 - 0,0138)^2 + (0,0181 - 0,0347)^2 + \dots + (0,0750 - 0,0750)^2} = 0,0835$$

$$D_2^+ = \sqrt{(y_2^+ - y_{2j})^2}$$

$$= \sqrt{(0,0564 - 0,0160)^2 + (0,0181 - 0,0361)^2 + \dots + (0,075 - 0,0231)^2} = 0,0865$$

$$\vdots$$

$$D_7^+ = \sqrt{(y_7^+ - y_{7j}^-)^2}$$

$$= \sqrt{(0,0564 - 0,0522)^2 + (0,0181 - 0,0236)^2 + \dots + (0,075 - 0,0281)^2} = 0,052$$

Persamaan (2.6) $D_i^- = \sqrt{\sum_{i=1}^n (y_{ij} - y_i^-)^2}$.

$$D_1^- = \sqrt{(y_{1j} - y_1^-)^2}$$

$$= \sqrt{(0,0138 - 0,0085)^2 + (0,0347 - 0,0403)^2 + \dots + (0,0754 - 0,0212)^2} = 0,0640$$

$$D_2^- = \sqrt{(y_{2j} - y_2^-)^2}$$

$$= \sqrt{(0,0160 - 0,0085)^2 + (0,0361 - 0,0403)^2 + \dots + (0,0441 - 0,0212)^2} = 0,0925$$

$$\vdots$$

$$D_7^- = \sqrt{(y_{7j} - y_7^-)^2}$$

$$= \sqrt{(0,0522 - 0,0085)^2 + (0,0236 - 0,0403)^2 + \dots + (0,0281 - 0,0212)^2} = 0,1193$$

Sehingga D^+ dan D^- yang didapat yaitu:

Tabel 4. 12 Hasil Perhitungan D⁺ dan D⁻

| D ⁺ | D ⁻ |
|----------------|----------------|
| 0,0835 | 0,0640 |
| 0,0865 | 0,0925 |
| 0,0845 | 0,0785 |
| 0,0734 | 0,0961 |
| 0,0544 | 0,1084 |
| 0,0761 | 0,0816 |
| 0,0924 | 0,0792 |
| 0,0836 | 0,0853 |
| 0,0861 | 0,0825 |
| 0,0522 | 0,1193 |

8. Menentukan Preferensi Untuk Setiap kriteria

Persamaan (2.7) digunakan untuk menghitung preferensi atau hasil akhir.

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}, V_1 = \frac{D_1^-}{D_1^- + D_1^+} = \frac{0,0640}{0,0640 + 0,0835} = 0,4338$$

$$V_2 = \frac{D_2^-}{D_2^- + D_2^+} = \frac{0,0925}{0,0925 + 0,0865} = 0,5168$$

⋮

$$V_7 = \frac{D_7^-}{D_7^- + D_7^+} = \frac{0,1193}{0,1193 + 0,0522} = 0,6957$$

Jadi Hasil akhir atau preferensi yang didapat yaitu:

Tabel 4. 13 Preferensi Untuk Setiap Kriteria

| |
|--------|
| V |
| 0,4338 |
| 0,5168 |
| 0,4817 |
| 0,5671 |
| 0,6660 |
| 0,5174 |
| 0,4616 |
| 0,5051 |
| 0,4892 |
| 0,6957 |

Setelah preferensi setiap kriteria dihitung maka didapatkan alternatif terbaik berupa *player*/alternatif dengan hasil akhir sebesar 0,6957 yaitu Xin.

4.1.3 Pemrograman Perhitungan Metode TOPSIS pada Python

Pada pemrograman perhitungan metode TOPSIS pada aplikasi python, skrip ditulis di menu *Jupyter Notebook* yang merupakan salah satu menu pilihan python. Hal ini dikarenakan pada menu *Jupyter Notebook* kita bisa melihat hasil setiap langkah perhitungan untuk memastikan hasil perhitungan dari skrip yang sudah diinputkan sesuai dengan hasil perhitungan manual. Adapun langkah-langkah pemrograman pada menu *Jupyter Notebook* yaitu:

1. Tahap Menginput Matriks Data

Memasukkan alternatif, kriteria dan nilai data pada menu *Jupyter Notebook* aplikasi python dengan menginput skrip berikut:

```

alternatif = ["BAJAN", "LUMINAIRE", "REKT", "REXXY", "WANN", "
LEMON", "LJ", "R7", "VYN", "XIN"]
kriteria = ["KPG", "DPG", "APG", "GPM", "KP%", "DPM", "TANKED"
]
alternatifkriteria = [
    [1.3,2.5,6.7,545,62,1413,6461],
    [1.5,2.6,7.7,545,75,2509,1975],
    [1.7,2.4,3.6,668,40,1906,4770],
    [2.9,1.8,5.9,578,69,3360,1818],
    [5.3,2.3,3.7,849,69,3627,2971],
    [2.4,2.3,4.3,690,52,1901,4324],
    [0.8,2.9,6.0,564,54,1138,6502],
    [3.4,1.3,2.8,706,45,1636,3427],
    [1.7,2.5,5.8,563,54,2353,2474],
    [4.9,1.7,4.5,892,70,4820,2408]
]

```

Selanjutnya menginput ketujuh bobot kepentingan kriteria yang telah ditentukan pada menu *Jupyter Notebook* aplikasi python dengan menginput skrip berikut:

```

kepentingan = [0.1,0.1,0.1,0.2,0.2,0.15,0.15]

```

Kriteria *benefit* dan *cost* ditambahkan pada matriks kriteria dengan skrip:

```

costbenefit = ["benefit", "cost", "benefit", "benefit", "benefit", "benefit", "benefit"]

```

2. Tahap Perhitungan Metode TOPSIS

Langkah pertama pada tahap perhitungan yaitu mencari matriks ternormalisasi. Sebelum menghitung matriks ternormalisasi kita mencari matriks pembagi untuk mempermudah perhitungan normalisasi matriks. Untuk menghitung matriks pembagi (persamaan 2.1) pada aplikasi python menggunakan skrip berikut:

```

pembagi = [ ]
for i in range(len(kriteria)):
    pembagi.append(0)
for j in range(len(alternatif)):

```

```

    pembagi[i] = pembagi[i] + (alternatifkriteria[j][i]**2)
    pembagi[i] = pembagi[i]**(1/2)
print (pembagi)

```

Setelah skrip di atas di-*run* maka akan keluar hasil perhitungan matriks pembagi menggunakan aplikasi python yaitu:

```

[9.39095309327014, 7.199305522062527, 16.78868666691948, 2121.420
2789640717, 189.87364219396014, 8527.811266673296, 12851.71039200
6195]

```

Gambar 4. 1 *Output* Matriks Pembagi pada *Jupyter Notebook*

Untuk menghitung matriks ternormalisasi pada menu *Jupyter Notebook* aplikasi python dengan menginput skrip berikut:

```

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(normalisasi)

```

Setelah di-*run* maka hasil perhitungan matriks ternormalisasinya yaitu:


```
[[0.13843110354066426, 0.3472557168658367, 0.39907826817696923,
0.25690336111340156, 0.32653294729905497, 0.165693160391812, 0.50
27346402093501], [0.15972819639307415, 0.36114594554047025, 0.458
6421888003974, 0.25690336111340156, 0.39499953302305035, 0.294213
82832488063, 0.15367604309138932], [0.18102528924548403, 0.333365
48819120326, 0.21443011424434166, 0.3148833857316555, 0.210666417
61229352, 0.22350400828506278, 0.37115682306122894], [0.308807846
35994335, 0.2500241161434025, 0.35142713167822665, 0.272458977474
3965, 0.3633995703812063, 0.3940049673860498, 0.141459770298808],
[0.564372960588862, 0.3194752595165698, 0.2203865063066845, 0.400
203584560143, 0.3633995703812063, 0.4253142906872627, 0.231175455
20228744], [0.25556511422891864, 0.3194752595165698, 0.2561248586
807414, 0.32525379663898546, 0.27386634289598155, 0.2229176913693
0972, 0.33645327105173034], [0.08518837140963954, 0.4028166315643
7064, 0.3573835237405694, 0.2658596250788229, 0.2843996637765962
6, 0.13344573002539425, 0.5059248770532726], [0.3620505784909680
6, 0.18057297277023512, 0.16677897774559905, 0.33279591366249817,
0.2369997198138302, 0.19184289483439806, 0.2666571137590842], [0.
18102528924548403, 0.3472557168658367, 0.34547073961588376, 0.265
38824276485334, 0.28439966377659626, 0.27592074055338545, 0.19250
35598015682], [0.5217787748840422, 0.23613388746876898, 0.2680376
428054271, 0.42047302406083337, 0.3686662308215136, 0.56520950678
59405, 0.18736805658940026]]
```

Gambar 4. 2 Output Matriks Ternormalisasi pada *Jupyter Notebook*

Setelah didapatkan matriks ternormalisasi maka selanjutnya kita memberi bobot matriks tersebut atau dapat kita sebut matriks ternormalisasi terbobot. Untuk menghitung matriks ternormalisasi terbobot (persamaan 2.2) pada menu *Jupyter Notebook* aplikasi python dengan menginput skrip berikut:

```
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(terbobot)
```

Setelah skrip di atas di-*run* maka akan keluar hasil perhitungan matriks ternormalisasi terbobot menggunakan aplikasi Python yaitu:


```
[[0.013843110354066428, 0.034725571686583676, 0.03990782681769692
4, 0.051380672222680314, 0.06530658945981099, 0.0248539740587718,
0.07541019603140252], [0.015972819639307415, 0.03611459455404702,
0.04586421888003975, 0.051380672222680314, 0.07899990660461008,
0.04413207424873209, 0.023051406463708398], [0.01810252892454840
3, 0.03333654881912033, 0.021443011424434166, 0.0629766771463311,
0.04213328352245871, 0.03352560124275942, 0.05567352345918434],
[0.030880784635994335, 0.025002411614340248, 0.03514271316782266
5, 0.05449179549487931, 0.07267991407624126, 0.05910074510790746
4, 0.021218965544821198], [0.0564372960588862, 0.0319475259516569
8, 0.02203865063066845, 0.0800407169120286, 0.07267991407624126,
0.0637971436030894, 0.034676318280343114], [0.025556511422891864,
0.03194752595165698, 0.025612485868074145, 0.0650507593277971, 0.
054773268579196316, 0.033437653705396454, 0.05046799065775955],
[0.008518837140963955, 0.04028166315643707, 0.035738352374056946,
0.05317192501576458, 0.05687993275531925, 0.020016859503809138,
0.07588873155799089], [0.036205057849096806, 0.01805729727702351,
0.016677897774559906, 0.06655918273249964, 0.04739994396276604,
0.028776434225159707, 0.03999856706386263], [0.01810252892454840
3, 0.034725571686583676, 0.034547073961588376, 0.0530776485529706
74, 0.05687993275531925, 0.041388111083007814, 0.0288755339702352
27], [0.052177877488404224, 0.0236133887468769, 0.026803764280542
708, 0.08409460481216668, 0.07373324616430273, 0.0847814260178910
7, 0.028105208488410038]]
```

Gambar 4. 3 *Output* Matriks Ternormalisasi Terbobot pada *Jupyter Notebook*

Selanjutnya yaitu mencari jarak ideal positif dan negatif. Untuk menghitung solusi ideal positif A^+ (persamaan 2.3) pada menu *Jupyter Notebook* aplikasi python dengan menginput skrip berikut:

```
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]
            elif aplus[i] > terbobot[j][i]:
                aplus[i] = terbobot[j][i]
    else:
        for j in range(len(alternatif)):
            if j == 0:
                aplus[i] = terbobot[j][i]
            elif aplus[i] < terbobot[j][i]:
                aplus[i] = terbobot[j][i]
print (aplus)
```

Selanjutnya skrip untuk menghitung solusi ideal negatif A^- (persamaan 2.4)

pada menu *Jupyter Notebook* aplikasi python dengan menginput skrip berikut:

```
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] = terbobot[j][i]
            elif amin[i] < terbobot[j][i]:
                amin[i] = terbobot[j][i]
    else:
        for j in range(len(alternatif)):
            if j == 0:
                amin[i] = terbobot[j][i]
            elif amin[i] > terbobot[j][i]:
                amin[i] = terbobot[j][i]
print(amin)
```

Setelah kedua skrip di atas di-*run* maka akan keluar hasil perhitungan solusi ideal positif A^+ dan solusi ideal negatif A^- pada menu *Jupyter Notebook* aplikasi python dengan menginput skrip berikut:

```
[0.0564372960588862, 0.01805729727702351, 0.04586421888003975, 0.08409460481216668, 0.07899990660461008, 0.08478142601789107, 0.07588873155799089]
```

Gambar 4. 4 *Output* Solusi Ideal positif A^+ pada *Jupyter Notebook*

```
[0.008518837140963955, 0.04028166315643707, 0.016677897774559906, 0.051380672222680314, 0.04213328352245871, 0.020016859503809138, 0.021218965544821198]
```

Gambar 4. 5 *Output* Solusi Ideal Negatif A^- pada *Jupyter Notebook*

Selanjutnya yaitu menentukan jarak antara nilai terbobot setiap alternatif terhadap jarak positif dan negatif. Untuk menghitung jarak antara setiap alternatif terhadap solusi ideal positif D^+ (persamaan 2.5) pada menu *Jupyter*

Notebook aplikasi python dengan menginput skrip berikut

```
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)
print(dplus)
```

Sedangkan untuk menghitung jarak antara setiap alternatif terhadap solusi ideal negatif D^- (persamaan 2.6) pada menu *Jupyter Notebook* aplikasi python dengan menginput skrip berikut:

```
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]) * (ter
bobot[i][j] - amin[j]))
        dmin[i] = dmin[i]**(1/2)
print(dmin)
```

Setelah skrip di atas di-*run* maka akan keluar hasil perhitungan jarak antara nilai terbobot setiap alternatif terhadap solusi ideal negatif D^+ dan solusi ideal negatif D^- :

```
[0.08352730959697956, 0.08647423861880849, 0.08450049852372607,
0.07335441286639369, 0.05436704502119254, 0.07608343490141045, 0.
09237201286617444, 0.08361756084846858, 0.08613958938422588, 0.05
218540521691956]
```

Gambar 4. 6 *Output D^+* pada *Jupyter Notebook*

```
[0.06399950693522613, 0.053561338434464006, 0.040827173109224194,
0.059538117799295444, 0.07903595787644793, 0.042682909261677816,
0.059772539645224804, 0.04413590071409653, 0.034317948432332435,
0.09272057371705196]
```

Gambar 4. 7 Output D⁻ pada Jupyter Notebook

Langkah terakhir tahap perhitungan yaitu mencari nilai akhir atau nilai preferensi dari setiap kriteria. Untuk menghitung preferensi dari setiap kriteria (persamaan 2.7) pada aplikasi python menggunakan skrip berikut :

```
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(alternatif)
print(hasil)
```

Setelah skrip di atas di-*run* maka akan keluar hasil preferensi dari setiap kriteria:

```
['BAJAN', 'LUMINAIRE', 'REKT', 'REXXY', 'WANN', 'LEMON', 'LJ', 'R
7', 'VYN', 'XIN']
[0.433816091471443, 0.38248379134459626, 0.32576343737395497, 0.4
480170367819423, 0.5924601107899489, 0.3593855613090713, 0.392866
7747781927, 0.3454771414744149, 0.28489664536057807, 0.6398671359
123242]
```

Gambar 4. 8 Output Hasil Preferensi pada Jupyter Notebook

3. Tahap Perangkingan

Untuk mengurutkan alternatif sehingga didapat perangkingan alternatif yang memiliki kriteria dari yang tertinggi hingga terendah, maka dibutuhkan skrip perangkingan berikut:

```
alternatifrangking = []
hasilrangking = []
```



```

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

```

Setelah skrip di atas di-run maka didapatkan perangkingan alternatif dari alternatif tertinggi ke terendah.

```

['XIN', 'WANN', 'REXXY', 'BAJAN', 'LJ', 'LUMINAIRE', 'LEMON', 'R
7', 'REKT', 'VYN']

```

Gambar 4. 9 Output Hasil Perangkingan Alternatif pada *Jupyter Notebook*

4.1.4 Aplikasi Metode TOPSIS pada Python

Pada aplikasi metode TOPSIS pada python ini terdapat tiga langkah penyelesaian. Langkah pertama yaitu tahapan input variabel data. Langkah kedua yaitu input nilai data dalam artian input nilai atribut sesuai dengan kriteria masing-masing. Langkah ketiga yaitu *ouput* yang berupa hasil akhir perhitungan serta perangkingan alternatif dengan urutan alternatif tertinggi ke alternatif terendah.

Sebelum memasuki langkah-langkah dalam aplikasinya, dibutuhkan skrip pembangkit untuk memanggil/menjalankan aplikasi metode TOPSIS pada Python.

Skrip yang digunakan yaitu:

```

import numpy as np
alternatif = str(input("Masukkan alternatif : "))

```

```

alternatif = alternatif.split(", ")
kriteria = str(input("Masukkan kriteria : "))
kriteria = kriteria.split(", ")
costbenefit = str(input("Masukkan cost/benefit : "))
costbenefit = costbenefit.split(", ")
kepentingan = str(input("Masukkan bobot kepentingan : "))
kepentingan = kepentingan.split(", ")
alternatifkriteria = list(np.zeros(len(alternatif)))
for i in range(len(alternatif)):
    alternatifkriteria[i] = str(input("Masukkan nilai data : "
))
    alternatifkriteria[i] = alternatifkriteria[i].split(", ")

```

Untuk skrip lengkap dapat dilihat pada lampiran 2.

Setelah skrip di-*run* maka aplikasi yang sudah dibuat akan berjalan dan akan muncul tampilan-tampilan perintah dan output hasil perhitungannya.

1. Input Variabel Data

```

===== RESTART: E:\Aplikasi.py =====
Masukkan alternatif : BAJAN, LUMINAIRE, REKT, REXXY, WANN, LEMON, LJ, R7, VYN, XIN
Masukkan kriteria : KPG, DPG, APG, GPM, KP%, DPM, TANKED
Masukkan cost/benefit : benefit, cost, benefit, benefit, benefit, benefit, benefit
Masukkan bobot kepentingan : 0.1, 0.1, 0.1, 0.2, 0.2, 0.15, 0.15

```

Gambar 4. 10 Tahap Input Variabel Data pada Aplikasi

Tahapan input variabel data meliputi atribut, kriteria, *cost/benefit* dan bobot kepentingan kriteria alternatif. Pada bagian alternatif berisikan nama-nama 10 *player*. Pada bagian kriteria berisikan 7 atribut yang digunakan untuk menentukan MVP. Pada bagian *cost/benefit* menentukan apakah kriteria termasuk *cost/benefit*. Pada bagian bobot berisikan bobot masing-masing kriteria.

2. Input Nilai Data

```
Masukkan nilai data : 1.3, 2.5, 6.7, 545, 62, 1413, 6461
Masukkan nilai data : 1.5, 2.6, 7.7, 545, 75, 2509, 1975
Masukkan nilai data : 1.7, 2.4, 3.6, 668, 40, 1906, 4770
Masukkan nilai data : 2.9, 1.8, 5.9, 578, 69, 3360, 1818
Masukkan nilai data : 5.3, 2.3, 3.7, 849, 69, 3627, 2971
Masukkan nilai data : 2.4, 2.3, 4.3, 690, 52, 1901, 4324
Masukkan nilai data : 0.8, 2.9, 6.0, 564, 54, 1138, 6502
Masukkan nilai data : 3.4, 1.3, 2.8, 706, 45, 1636, 3427
Masukkan nilai data : 1.7, 2.5, 5.8, 563, 54, 2353, 2474
Masukkan nilai data : 4.9, 1.7, 4.5, 892, 70, 4820, 2408
```

Gambar 4. 11 Tahap Input Nilai Data

Tahapan input nilai data yaitu menginput nilai data sesuai dengan kriteria dan alternatifnya. Pada data kali ini input data berbentuk matriks 10×7 karena datanya terdiri dari 10 alternatif dan 7 kriteria.

3. Hasil Akhir dan Perangkingan

```
alternatif = ['BAJAN', 'LUMINAIRE', 'REKT', 'REXXY', 'WANN', 'LEMON', 'LJ', 'R7', 'VYN',
             'XIN']
hasil = [0.433816091471443, 0.38248379134459626, 0.32576343737395497, 0.4480170367819423
, 0.5924601107899489, 0.3593855613090713, 0.3928667747781927, 0.3454771414744149, 0.28489
664536057807, 0.6398671359123242]
Rangking alternatif = ['XIN', 'WANN', 'REXXY', 'BAJAN', 'LJ', 'LUMINAIRE', 'LEMON', 'R7',
, 'REKT', 'VYN']
```

Gambar 4. 12 Hasil Akhir dan Perangkingan

Setelah menginput nilai data maka akan keluar hasil akhir serta perangkingan alternatif. Pada baris pertama ditampilkan lagi alternatif 10 pemain yang sudah diinput di awal. Pada baris kedua ditampilkan hasil perhitungan alternatif kriteria menggunakan metode TOPSIS. Pada baris ketiga ditampilkan hasil perangkingan alternatif yang mana urutan alternatif dari yang terbaik ke terendah yaitu: XIN, WANN, REXXY, BAJAN, LJ, LUMINAIRE, LEMON, LJ, R7, REKT dan VYN.

4.2 Pembahasan

4.2.1 Hasil Perhitungan Manual Metode TOPSIS

Hasil perhitungan manual metode TOPSIS dalam menentukan *Most Valuable*

Player yaitu:

Tabel 4. 14 Hasil Perhitungan Manual Metode TOPSIS

| <i>Player</i> | Hasil Akhir Perhitungan |
|---------------|-------------------------|
| BAJAN | 0,4338 |
| LUMINAIRE | 0,5168 |
| R E K T | 0,4817 |
| REXXY | 0,5671 |
| WANN | 0,6660 |
| LEMON | 0,5174 |
| LJ | 0,4616 |
| R7 | 0,5051 |
| VYN | 0,4892 |
| XIN | 0,6957 |

Karena Xin memiliki hasil akhir terbesar yaitu 0,6957 maka *Most Valuable Player* dari *MPL ID* season 5 adalah XIN.

4.2.2 Hasil Perhitungan Metode TOPSIS Pada Python

Hasil perhitungan pada pemrograman metode TOPSIS pada Python untuk menentukan *Most Valuable Player* yaitu: XIN, WANN, REXXY, BAJAN, LJ, LUMINAIRE, LEMON, R7, REKT, VYN. Hasil akhir tersebut sudah terurut dari *player* yang memiliki hasil akhir tertinggi hingga yang terendah. Sehingga *Most Valuable Player* dari MPL ID season 5 adalah XIN.

Sedangkan menggunakan aplikasi metode TOPSIS pada Python untuk menentukan *Most Valuable Player output* yang dihasilkan pada hasil akhir perangkingan yaitu: XIN, WANN, REXXY, BAJAN, LJ, LUMINAIRE, LEMON, R7, REKT, VYN. Hasil akhir tersebut sudah terurut dari *player* yang memiliki hasil akhir tertinggi hingga yang terendah. Sehingga *Most Valuable Player* dari MPL ID season 5 adalah XIN.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah diuraikan mengenai metode TOPSIS dan penerapannya di python, maka dapat diambil kesimpulan sebagai berikut:

1. Sebelum membuat pemrograman dan aplikasi metode TOPSIS pada python, diperlukan perhitungan metode TOPSIS secara manual yang digunakan sebagai acuan dalam membuat algoritma pemrograman dan aplikasinya.

Adapun tahap-tahapnya yaitu:

- Tahap 1. Tahap menginput semua nilai data. Tahap ini yaitu menginput matriks data yang berisikan alternatif, kriteria dan nilai data. Setelah data dan nilainya dimasukkan langkah selanjutnya yaitu memberi kriteria *cost/benefit* dan bobot kepentingan pada setiap kriteria.
- Tahap 2. Tahap perhitungan metode TOPSIS. Langkah pertama dalam tahap ini yaitu membuat matriks ternormalisasi yang mana dibutuhkan matriks pembagi untuk membantu perhitungan matriks ternormalisasi. Setelah didapatkan matriks ternormalisasi selanjutnya yaitu memberi bobot terhadap setiap elemen matriks sesuai dengan bobot kriterianya sehingga didapatkan matriks ternormalisasi terbobot. Langkah berikutnya yaitu mencari jarak ideal positif A^+ dan jarak ideal negatif A^- . Setelah mendapatkan nilai jarak ideal positif dan negatif, dicari jarak antara nilai terbobot setiap alternatif terhadap jarak positif D^+ dan negatif D^- . Langkah terakhir tahap perhitungan yaitu mencari nilai akhir

atau nilai preferensi dari setiap kriteria yang merupakan hasil perhitungan dari metode TOPSIS.

- Tahap 3. Tahap terakhir metode TOPSIS yaitu meranking alternatif berdasarkan hasil akhir yang sudah dihitung dengan cara mengurutkan dari yang terbesar hingga terkecil.
2. Hasil perhitungan metode TOPSIS dalam menentukan *Most Valuable Player Mobile Legends Professional League Indonesia season 5* adalah Bajan dengan nilai akhir 0,4338, LUMINAIRE dengan hasil akhir 0,3825, REKT dengan hasil akhir 0,3258, REXXY dengan hasil akhir 0,4481, WANN dengan hasil akhir 0,5925, LEMON dengan hasil akhir 0,3594, LJ dengan hasil akhir 0,3929, R7 dengan hasil akhir 0,3455, VYN dengan hasil akhir 0,2849 dan XIN dengan hasil akhir 0,6398. Karena XIN memiliki nilai akhir tertinggi yaitu 0,6398 maka XIN adalah *Most Valuable Player Mobile Legends Professional League Indonesia season 5*.

5.2 Saran

Dalam skripsi ini metode TOPSIS hanya digunakan untuk mencari alternatif dan kriteria yang jumlahnya sedikit serta aplikasi yang dibuat hanya dalam bahasa pemrograman python. Oleh karena itu, penulis menyarankan untuk:

1. Menggunakan metode TOPSIS untuk menyelesaikan permasalahan dengan jumlah alternatif yang sangat besar.

2. Membuat aplikasi sejenis dengan menggunakan bahasa pemrograman yang lainnya seperti *java*, *php*, *r*, *exe* dan lain sebagainya.



DAFTAR RUJUKAN

- Alamsyah, M. N. (2017). Implementasi Metode TOPSIS Dalam Memprediksi Tingkat Kelulusan Mahasiswa Di STMIK Mura Muhammad. *JTI*, 9(2), 87–97. <https://doi.org/10.1017/CBO9781107415324.004>
- Azizi, F. N. (2014). *Implementasi Metode TOPSIS Pada Sistem Pendukung Keputusan Untuk Seleksi Siswa Kelas Unggulan*. UIN Maulana Malik Ibrahim Malang.
- Efenie, Y., & Hozairi. (2016). Implementasi Metode TOPSIS Untuk Analisis Keberhasilan Pengelolaan Sekolah Berbasis Web. *Jurnal MNEMONIC*, 2(2), 32.
- Harismawan, A. F., Kharisma, A. P., & Afirianto, T. (2018). Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python , PHP , dan Perl pada Client Berbasis Android. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 2(January), 237–245. <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/781>
- Haristu, R. A. (2019). Penerapan Metode *Random Forest* Untuk Prediksi *Win Ratio* Pemain *Player Unknown Battleground*. Universitas Sanata Dharma.
- Hidayat, L. N. (2014). Metode TOPSIS Untuk Membantu Pemilihan Jurusan Pada Sekolah Menengah Atas. *Jurnal Universitas Dian Nuswantoro*, 3, 5–6.
- Karmila, Ridwan, M., Parlina, I., & Satria, H. (2017). Sistem Pendukung Keputusan dalam Merekomendasikan Smartphone untuk Kalangan Pemula dengan Metode TOPSIS. *Seminar Nasional Multidisiplin Ilmu 2017*.
- Saragih, R. R. (2018). Pemrograman dan bahasa pemrograman. December. <https://www.researchgate.net/publication>
- Sari, D. R., Windarto, A. P., Hartama, D., & Solikhun, S. (2018). Decision Support System for Thesis Graduation Recommendation Using AHP-TOPSIS Method. *Jurnal Teknologi Dan Sistem Komputer*, 6(1), 1–6. <https://doi.org/10.14710/jtsiskom.6.1.2018.1-6>
- Sunarti. (2018). Perbandingan Metode TOPSIS dan SAW Untuk Pemilihan Rumah Tinggal. *Journal of Information System*, 3(1), 69–79. <https://publikasi.dinus.ac.id/index.php/joins/article/view/1883/1289>

LAMPIRAN-LAMPIRAN

Lampiran 1. Data Atribut Player Selama Gelaran *Playoffs MPL ID season 5*

| <i>PLAYER</i> | <i>GAMES</i> | <i>KPG</i> | <i>DPG</i> | <i>APG</i> | <i>KDA</i> | <i>GPM</i> | <i>KP%</i> | <i>DPM</i> | <i>TANKED</i> | <i>KILLS</i> | <i>DEATHS</i> | <i>ASSISTS</i> |
|---------------|--------------|------------|------------|------------|------------|------------|------------|------------|---------------|--------------|---------------|----------------|
| BAJAN | 15 | 1.3 | 2.5 | 6.7 | 3.24 | 545 | 62% | 1413 | 6461 | 19 | 37 | 101 |
| LUMINAIRE | 15 | 1.5 | 2.6 | 7.7 | 3.54 | 545 | 75% | 2509 | 1975 | 22 | 39 | 116 |
| REKT | 15 | 1.7 | 2.4 | 3.6 | 2.22 | 668 | 40% | 1906 | 4770 | 26 | 36 | 54 |
| REXXY | 15 | 2.9 | 1.8 | 5.9 | 4.89 | 578 | 69% | 3360 | 1818 | 44 | 27 | 88 |
| WANN | 15 | 5.3 | 2.3 | 3.7 | 3.97 | 849 | 69% | 3627 | 2971 | 80 | 34 | 55 |
| LEMON | 10 | 2.4 | 2.3 | 4.3 | 2.91 | 690 | 52% | 1901 | 4324 | 24 | 23 | 43 |
| LJ | 10 | 0.8 | 2.9 | 6 | 2.34 | 564 | 54% | 1138 | 6502 | 8 | 29 | 60 |
| R7 | 10 | 3.4 | 1.3 | 2.8 | 4.77 | 706 | 45% | 1636 | 3427 | 34 | 13 | 28 |
| VYN | 10 | 1.7 | 2.5 | 5.8 | 3 | 563 | 54% | 2353 | 2474 | 17 | 25 | 58 |
| XIN | 10 | 4.9 | 1.7 | 4.5 | 5.53 | 892 | 70% | 4820 | 2408 | 49 | 17 | 45 |
| ANTIMAGE | 9 | 2.2 | 2.6 | 3.1 | 2.09 | 618 | 44% | 2021 | 6029 | 20 | 23 | 28 |
| XW | 9 | 5.1 | 2.1 | 4.1 | 4.37 | 816 | 76% | 4726 | 2425 | 46 | 19 | 37 |
| SASA | 9 | 1.4 | 2.6 | 5.1 | 2.57 | 537 | 62% | 1384 | 3532 | 13 | 23 | 46 |
| RASY | 8 | 1.1 | 3.3 | 6.6 | 2.38 | 531 | 63% | 1373 | 6405 | 9 | 26 | 53 |
| UDIL | 7 | 1.7 | 2 | 7.6 | 4.64 | 541 | 74% | 2783 | 1638 | 12 | 14 | 53 |
| BRANZ | 5 | 4.4 | 2.4 | 3 | 3.08 | 752 | 80% | 3475 | 2688 | 22 | 12 | 15 |
| BRAVO | 5 | 2.6 | 1.4 | 3.6 | 4.43 | 588 | 65% | 1265 | 3368 | 13 | 7 | 18 |
| DREAMS | 5 | 0,8 | 3 | 5,8 | 2,2 | 525 | 74% | 2091 | 1198 | 4 | 15 | 29 |
| KYY | 5 | 0.8 | 4 | 5 | 1.45 | 513 | 64% | 1422 | 5905 | 4 | 20 | 25 |

| | | | | | | | | | | | | |
|-----------|---|-----|-----|-----|------|-----|------|------|-------|----|----|----|
| ANNISA | 3 | 1 | 1.7 | 5.3 | 3.8 | 547 | 60% | 1603 | 3576 | 3 | 5 | 16 |
| CAESIUS | 3 | 1.7 | 4 | 7.3 | 2.25 | 544 | 76% | 1995 | 3978 | 5 | 12 | 22 |
| CELIBOY | 3 | 6 | 1.7 | 6 | 7.2 | 852 | 100% | 4323 | 2880 | 18 | 5 | 18 |
| LEOMURPHY | 3 | 0.3 | 2.3 | 7.7 | 3.43 | 564 | 65% | 1864 | 3561 | 1 | 7 | 23 |
| MAUNGZY | 3 | 2 | 3 | 4 | 2 | 638 | 48% | 2018 | 5645 | 6 | 9 | 12 |
| RMITCHI | 3 | 2 | 4.3 | 5 | 1.62 | 604 | 59% | 1907 | 5481 | 6 | 13 | 15 |
| CLAY | 2 | 4 | 2 | 8 | 6 | 530 | 79% | 4042 | 1969 | 8 | 4 | 16 |
| JACKLEE | 2 | 1.5 | 4.5 | 9.5 | 2.44 | 539 | 74% | 1984 | 3612 | 3 | 9 | 19 |
| KIDO | 2 | 1 | 4 | 6.5 | 1.88 | 529 | 50% | 963 | 11618 | 2 | 8 | 13 |
| RINAZMI | 2 | 2.5 | 3 | 5.5 | 2.67 | 646 | 53% | 1406 | 4890 | 5 | 6 | 11 |
| WARLORD | 2 | 0.5 | 3.5 | 3.5 | 1.14 | 489 | 40% | 2970 | 4804 | 1 | 7 | 7 |
| PSYCHOO | 2 | 1.5 | 3.5 | 2 | 1 | 460 | 70% | 1614 | 2238 | 3 | 7 | 4 |
| WATT | 2 | 6 | 2 | 5.5 | 5.75 | 804 | 77% | 3298 | 2708 | 12 | 4 | 11 |
| LORDGOV | 1 | 0 | 4 | 2 | 0.5 | 433 | 67% | 1483 | 6970 | 0 | 4 | 2 |

Lampiran 2. Skrip Lengkap Aplikasi TOPSIS pada Python

```
import numpy as np
alternatif = str(input("Masukkan alternatif : "))
alternatif = alternatif.split(", ")
kriteria = str(input("Masukkan kriteria : "))
kriteria = kriteria.split(", ")
costbenefit = str(input("Masukkan cost/benefit : "))
costbenefit = costbenefit.split(", ")
kepentingan = str(input("Masukkan bobot kepentingan : "))
kepentingan = kepentingan.split(", ")
alternatifkriteria = list(np.zeros(len(alternatif)))
for i in range(len(alternatif)):
    alternatifkriteria[i] = str(input("Masukkan nilai data : "))
    alternatifkriteria[i] = alternatifkriteria[i].split(", ")

pembagi = []
for i in range(len(kriteria)):
    pembagi.append(0)
    for j in range(len(alternatif)):
        pembagi[i] = pembagi[i] +
        (float(alternatifkriteria[j][i])**2)
    pembagi[i] = pembagi[i]**(1/2)
#print(pembagi)

normalisasi = []
for i in range(len(alternatif)):
    normalisasi.append([])
    for j in range(len(kriteria)):
        normalisasi[i].append(0)
        normalisasi[i][j] = float(alternatifkriteria[i][j])/
pembagi[j]
#print(normalisasi)

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] *
float(kepentingan[j])
#print(terbobot)

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]
            elif aplus[i] > terbobot[j][i]:
                aplus[i] = terbobot[j][i]
```

```

else:
    for j in range(len(alternatif)):
        if j == 0:
            aplus[i] = terbobot[j][i]
        elif aplus[i] < terbobot[j][i]:
            aplus[i] = terbobot[j][i]

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] = terbobot[j][i]
            elif amin[i] < terbobot[j][i]:
                amin[i] = terbobot[j][i]
        else:
            for j in range(len(alternatif)):
                if j == 0:
                    amin[i] = terbobot[j][i]
                elif amin[i] > terbobot[j][i]:
                    amin[i] = terbobot[j][i]

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)
#print(dplus)

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)
#print(dmin)

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("alternatif = ", alternatif)
print("hasil = ", hasil)
alternatifrangking = []
hasilrangking = []

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

```



```
for i in range(len(alternatif)): # Algoritma Sorting
    for j in range(len(alternatif)):
        if j > i:
            if hasilranking[j] > hasilranking[i]:
                tmphasil = hasilranking[i]
                tmpalternatif = alternatiffranking[i]
                hasilranking[i] = hasilranking[j]
                alternatiffranking[i] = alternatiffranking[j]
                hasilranking[j] = tmphasil
                alternatiffranking[j] = tmpalternatif
print("Rangking alternatif = ", alternatiffranking)
```



RIWAYAT HIDUP



Mohammad Rifqi Riza Pahlevy lahir di Kabupaten Pasuruan pada tanggal 26 April 1998. Memiliki nama panggilan Rifqi. Beralamat Dusun Bajangan Barat RT/RW 002/001 Desa Bajangan Kec. Gondangweta Kab. Pasuruan. Merupakan anak kedua dari Bapak Akhmad Khusaeri dan Ibu Sunarti.

Pendidikan yang pernah ditempuh yaitu RA Miftahul Ulum Bajangan. Kemudian melanjutkan sekolahnya di Sekolah Dasar Negeri Warungdowo dan lulus pada tahun 2010. Menempuh pendidikan SMP di Sekolah Menengah Pertama Negeri 6 Pasuruan lulus pada tahun 2013. Melanjutkan pendidikan SMA di Sekolah Menengah Atas Negeri 1 Pasuruan lulus pada tahun 2016.

Tahun 2016 melanjutkan studi ke jenjang pendidikan strata 1 di Universitas Islam Negeri Maulana Malik Ibrahim Malang mengambil jurusan matematika Fakultas Sains dan Teknologi. Turut aktif mengikuti kegiatan organisasi atau komunitas yang ada di dalam dan di luar (intra atau ekstra) kampus.

Prestasi-prestasi yang pernah diraihinya yaitu juara 2 Olimpiade Sains Nasional (OSN) mata pelajaran matematika tingkat SMP se-Kota Pasuruan dan

juara 2 Liga Pendidikan Indonesia (LPI) tingkat SMP se-Kota Pasuruan. Keduanya diraih di tahun yang sama yaitu tahun 2012.





KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No. 50 Dinoyo Malang Telp./Fax.(0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Mohammad Rifqi Riza Pahlevy
NIM : 16610094
Fakultas/Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Aplikasi Metode TOPSIS Pada Python Untuk
Menentukan *Most Valuable Player* Turnamen *Mobile Legends Professional League Indonesia Season 5*.
Pembimbing I : Ari Kusumastuti, M.Si, M.Pd
Pembimbing II : Angga Dwi Mulyanto, M.Si

| No | Tanggal | Hal | Tanda Tangan |
|----|---------------|---|--------------|
| 1 | 20 April 2020 | Konsultasi Bab I, Bab II, Bab III dan Bab IV | 1. |
| 2 | 27 April 2020 | Konsultasi Kajian Keagamaan pada Bab I dan Bab II | 2. |
| 3 | 29 April 2020 | Revisi Bab I, Bab II, Bab III dan Bab IV | 3. |
| 4 | 1 Mei 2020 | ACC Bab I, Bab II, Bab III dan Bab IV | 4. |
| 5 | 20 Mei 2020 | Konsultasi Bab I, Bab IV dan Bab V | 5. |
| 6 | 25 Mei 2020 | Revisi Bab I, Bab IV dan Bab V | 6. |
| 7 | 1 Juni 2020 | ACC Bab I, Bab IV dan Bab V | 7. |
| 8 | 5 Juni 2020 | Konsultasi Keseluruhan | 8. |
| 9 | 10 Juni 2020 | Revisi Keseluruhan | 9. |
| 10 | 17 Juni 2020 | ACC Keseluruhan | 10. |

Malang, 17 Juni 2020

Mengetahui,

Ketua Jurusan Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001