

Implementasi *Analytic Hierarchy Process* (AHP) Dalam Pengambilan Keputusan Desain Kualitas *Software*

Muthiah As Saidah¹, Muhammad Qolbi Shobri², Nila Destia Nasra³

¹*Program Studi Sistem Informasi, Sekolah Tinggi Teknologi Indonesia Tanjung Pinang
Jalan Pompa Air No. 28, Tanjungpinang, Indonesia, 29122*

²*Program Studi Ilmu Aktuaria, Fakultas Ilmu Formal dan Ilmu Terapan Universitas Muhammadiyah Madiun
Jl. Mayjend Panjaitan No. 18 Banjarejo, Taman, Madiun, Indonesia*

³*Program Studi Sains Aktuaria, Universitas YPPI Rembang
Jl. Mayjend Panjaitan No. 18 Banjarejo, Taman, Madiun, Indonesia*

Korespondensi Email : muthiahassaidah40@gamil.com

²mqs151@ummad.ac.id

³iladestia.id@gmail.com

Intisari— Peningkatan kualitas *software* merupakan aspek krusial dalam rekayasa perangkat lunak, yang mengharuskan pemahaman mendalam terhadap struktur dan kualitas yang tersedia di pasar. Kualitas *software* melibatkan sejumlah karakteristik luar, seperti ketepatan, daya guna, efisiensi, keandalan, integritas, kemampuan adaptasi, keakuratan, dan kekuatan, yang dapat diamati langsung oleh pengguna. Oleh karena itu, *programmer* perlu memastikan bahwa perangkat lunak yang mereka rancang dapat memenuhi kriteria tersebut. Berkaitan dengan hal itu, penelitian ini dilakukan untuk melihat perspektif dari pengguna *software* tentang prioritas pengguna terhadap delapan kriteria karakteristik luar dari sebuah *software* dengan menggunakan metode *Analytic Hierarchy Process* (AHP). Penelitian ini merupakan penelitian kuantitatif yang bertujuan untuk menentukan prioritas yang akan pilih oleh *programmer* dalam memilih karakteristik luar saat merancang perangkat lunak/ *software*. Sampel yang diambil berjumlah 30 orang yang pernah dan sedang menggunakan perangkat lunak/ *software* yang dipilih menggunakan teknik *random sampling*. Data diperoleh dari penyebaran kuisioner dengan skala Saaty 1-9. Berdasarkan hasil penelitian nilai *consistency ratio* (CR) dari data adalah 0.07 artinya kriteria perbandingan berpasangan yang diberikan oleh pengguna *software* konsisten karena nilai $CR \leq 0.1$ dan diperoleh *eigen vektor* dari matriks perbandingan kriteria atau disebut juga prioritas pengguna terhadap karakteristik luar *software* yang akan digunakan dari delapan kriteria yang telah diberikan. Prioritas ini memberikan landasan yang kuat bagi *programmer* untuk fokus pada aspek-aspek yang dianggap paling penting oleh pengguna.

Kata kunci— *Analytic Hierarchy Process* (AHP), Karakteristik Luar, *Software*, Matriks Perbandingan Pasangan, *Eigen Vektor*, *Consistency Ratio*.

Abstract— The improvement of software quality is a crucial aspect in software engineering, requiring a profound understanding of the structure and quality available in the market. Software quality involves several external characteristics, such as accuracy, usability, efficiency, reliability, integrity, adaptability, accuracy, and strength, which can be directly observed by users. Therefore, programmers need to ensure that the software they design meets these criteria. In connection with this, this research was conducted to examine the user's perspective on their priorities regarding the eight external characteristic criteria of software using the Analytic Hierarchy Process (AHP) method. This quantitative study aims to determine the priorities chosen by programmers when selecting external characteristics during software design. A sample of 30 individuals who have used selected software was randomly selected for data collection through questionnaires with a Saaty scale of 1-9. Based on the research results, The consistency ratio (CR) of the data is 0.07, indicating that the pairwise comparison criteria provided by the user in the software are consistent because of the value $CR \leq 0.1$ and the eigen vector of the criteria comparison matrix, also known as the user's priority for external software characteristics, was obtained for the eight criteria provided. This priority provides a strong foundation for programmers to focus on aspects considered most important by users.

Keywords— *Analytic Hierarchy Process* (AHP), *External Characteristics*, *Software*, *Pairwise Comparison Matrix*, *Eigen Vektor*, *Consistency Ratio*

I. PENDAHULUAN

Peningkatan kualitas *software* memiliki peran yang sangat penting dalam praktek rekayasa perangkat lunak. Oleh karena itu, pengetahuan tentang kualitas *software* sangat diperlukan untuk memahami perbedaan struktur dan kualitas yang tersedia di pasar. Kualitas *software* artinya *software* memiliki

kesesuaian dengan spesifikasi dan memenuhi kebutuhan pengguna [1]. Menurut McConnell *software* memiliki karakteristik luar dan karakteristik dalam. Karakteristik luar atau disebut juga kualitas *software* meliputi karakteristik-karakteristik yang dapat diamati secara langsung oleh pengguna. Karakteristik luar memiliki banyak kriteria seperti ketepatan, daya guna, efisiensi, keandalan, integritas,

kemampuan adaptasi, keakuratan, dan kekuatan [13][2]. *Software* yang dirancang oleh *programmer* harus dapat memenuhi karakteristik tersebut [2]. Oleh karena itu perlu adanya suatu metode untuk melihat desain prioritas karakteristik sebuah *software* agar dapat memenuhi kebutuhan pengguna.

Analytic Hierarchy Process (AHP) merupakan sebuah metode yang digunakan untuk menyelesaikan masalah pengambilan keputusan dengan banyak kriteria (multicriteria) [3][10][14][15]. Metode *Analytic Hierarchy Process* (AHP) diperkenalkan oleh Thomas L. Saaty memiliki tiga prinsip utama yaitu prinsip hirarki, prinsip prioritas, dan prinsip konsistensi [9].

Analytic Hierarchy Process (AHP) memiliki kemampuan untuk mengatasi situasi yang kompleks, seperti kondisi yang tidak terstruktur, dengan menguraikan elemen-elemen hierarki menjadi komponen-komponen yang dinilai secara numerik. Pendekatan ini melibatkan perbandingan variabel dengan mempertimbangkan faktor-faktor subjektif, sehingga dapat menentukan prioritas tertinggi [5][2][6].

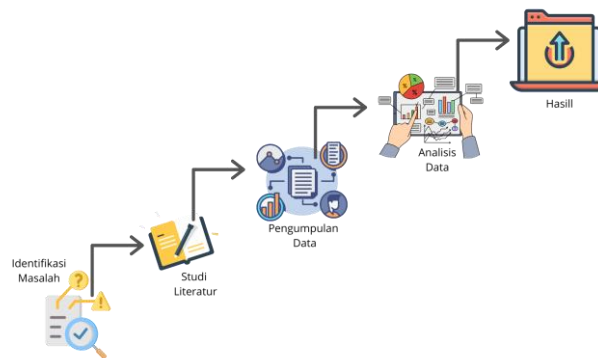
Beberapa penelitian telah banyak menggunakan metode *Analytic Hierarchy Process* (AHP) dalam melakukan pengambilan keputusan dengan permasalahan terkait dengan pemilihan prioritas dengan banyak kriteria. Berdasarkan penelitian [4] menggunakan AHP dalam pemilihan prioritas paket layanan internet. Demikian juga [6] penelitiannya telah mengeksplorasi sistem penunjang keputusan yang dikembangkan untuk memberikan rekomendasi *game* unit di pusat permainan, dengan tujuan memberikan pelayanan rekomendasi *gaming unit* yang sesuai dengan kebutuhan pelanggan.

Berkaitan dengan hal itu, seorang *programmer* juga memerlukan perspektif dari pengguna *software* tentang prioritas pengguna terhadap delapan kriteria karakteristik luar dari sebuah *software* dengan menggunakan metode *Analytic Hierarchy Process* (AHP). Delapan kriteria karakteristik luar tersebut yaitu ketepatan, daya guna, efisiensi, keandalan, integritas, kemampuan adaptasi, keakuratan, dan kekuatan. Metode pengambilan keputusan *Analytic Hierarchy Process* (AHP) diharapkan dapat membantu *programmer* dalam menentukan prioritas untuk mendesain karakteristik luar dalam meningkatkan kualitas *software*.

II. METODOLOGI PENELITIAN

Penelitian ini merupakan penelitian kuantitatif. Metodologi penelitian kuantitatif adalah pendekatan ilmiah yang digunakan untuk mengumpulkan, menganalisis, dan menginterpretasikan data berdasarkan angka-angka dan ukuran kuantitatif [12]. Dalam konteks ini, penelitian kuantitatif bertujuan untuk menentukan prioritas yang akan pilih oleh *programmer* dalam memilih karakteristik luar saat merancang perangkat lunak/ *software*.

Adapun alur penelitian merupakan rangkaian tahapan yang dirancang secara sistematis untuk mencapai tujuan penelitian yang telah ditetapkan. Berikut ini adalah alur penelitian kuantitatif [13].



Gambar 1. Alur Penelitian

Pengumpulan data pada penelitian ini melibatkan penyebaran kuesioner kepada responden, yang diminta untuk memberikan penilaian skor pada rentang 1-9 menggunakan skala *Analytical Hierarchy Process* (AHP) atau sering disebut skala perbandingan Saaty. Skala penilaian perbandingan Saaty dan tabel *random index* yang berguna untuk menghitung nilai rasio konsistensi kriteria [8].

TABEL I
SKALA DASAR PENILAIAN PERBANDINGAN PASANGAN ANALYTIC HIERARCHY PROCESS (AHP)- SKALA SAATY

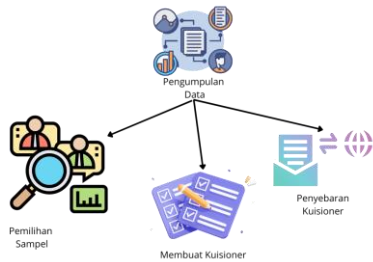
Tingkat Kepentingan	Definisi
1	Kedua elemen sama pentingnya (<i>Equal Importance</i>)
3	Elemen yang satu sedikit lebih penting dari pada elemen yang lainnya (<i>Moderate Importance</i>)
5	Elemen yang satu lebih penting dari pada elemen yang lainnya (<i>Strong Importance</i>)
7	Satu elemen jelas lebih mutlak penting dari pada elemen lainnya (<i>Very strong or demonstrated importance</i>)
9	Satu elemen mutlak penting dari pada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
1/c (Kebalikan)	Jika untuk aktivitas <i>i</i> mendapatkan satu angka dibanding dengan aktifitas <i>j</i> , maka <i>j</i> mempunyai nilai kebalikannya dibandingkan dengan <i>i</i>

TABEL II
RANDOM INDEKS

n	Random Indeks
1	0
2	0
3	0.58
4	0.9
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.54
13	1.56

Selanjutnya, sampel pada penelitian ini diambil berjumlah 30 orang yang pernah/ sedang menggunakan perangkat lunak/ *software*. Sampel dalam penelitian ini dipilih melalui teknik *random sampling*.

Berikut ini adalah tahapan dalam pengumpulan data penelitian



Gambar 2. Tahapan Pengumpulan Data

Adapun langkah- langkah *Analytic Hierarchy Process* (AHP) pada penelitian ini adalah sebagai berikut [3]:

1. Menentukan prioritas dari kombinasi karakteristik luar perangkat lunak/*software*.
2. Menentukan matriks perbandingan pasangan (*pairwise comparison matrix*)
3. Menghitung nilai-nilai eigen dan vektor eigen (prioritas) dari setiap kriteria- kriteria
4. Menghitung nilai eigen maksimal (λ_{max})
5. Menghitung nilai *Consistency Index* (CI) dan *Consistency Ratio* (CR) untuk mengetahui nilai rasio konsistensi kriteria dari karakteristik luar *software* [4] [11]:

$$CI = \frac{(\lambda_{max} - n)}{(n - 1)} \tag{1}$$

$$CR = \frac{CI}{IR} \tag{2}$$

6. *Consistency Ratio* $CR \leq 0.1$ menunjukkan hasil perhitungan kriteria konsisten. Perhitungan CR melibatkan rasio antara nilai konsistensi yang diukur dan nilai konsistensi acak yang diharapkan. Jika nilai CR mendekati nol, maka perbandingan berpasangan dianggap konsisten. Sebaliknya, jika nilai CR melebihi batas tertentu (biasanya 0.1), maka perbandingan berpasangan dianggap kurang konsisten, dan revisi mungkin diperlukan.

Klasifikasi parameter dan karakteristik pada penelitian ini terlihat pada tabel berikut :

TABEL III
PARAMETER KARAKTERISTIK LUAR SISTEM/PERANGKAT LUNAK

Karakteristik Luar Sistem/Perangkat Lunak	Parameter
Ketepatan	x_1
Daya Guna	x_2
Efisiensi	x_3
Keandalan	x_4
Integritas	x_5
Kemampuan Adaptasi	x_6
Keakuratan	x_7
Kekuatan	x_8

Karakteristik- karakteristik luar *softwar* pada tabel diatas mempunyai definisi sebagai berikut :

1. Ketepatan adalah tingkat keabsahan sistem dari kesalahan analisis, perancangan, serta implementasi.
2. Daya guna adalah tingkat kemudahan penggunaan untuk mempelajari dan menggunakan sitem.
3. Efisiensi artinya sebuah sistem meminimalisasi penggunaan memori, penggunaan sumberdaa CPU, penggunaan hardisk dan sebagainya.
4. Keandalan mencakup kemampuan *software* untuk menjalankan semua fungsi yang dibutuhkan dan diharapkan.
5. Intgritas adalah tingkat kemampuan *software* dalam mencegah akses yang tidak sah.
6. Kemampuan adaptasi artinya *software* dapat dipakai dalam aplikasi atau lingkungan yang lain dari lingkungan saat *software* diciptakan.
7. Kakuratan adalah tingkat kebebasan *software* dari kesalahan-kesalahan (*bugs*).
8. Kekuatan adalah tingkat kemampuan sistem melanjutkan fungsinya bila terdapat masukan yang tidak sah.

III. HASIL PENELITIAN

A. *Analytic Hierarchy Process* (AHP)

Berdasarkan langkah-langkah *Analytic Hierarchy Process* (AHP), langkah pertama yang harus dilakukan adalah menentukan prioritas dari kombinasi karakteristik luar perangkat lunak/*software*.

TABEL IV
PRIORITAS DARI KOMBINASI KARAKTERISTIK LUAR SISTEM/PERNGKAT LUNAK

Perbandingan Pasangan Karakteristik Luar Sistem/ Software	Aspek Paling Penting	Prioritas
$x_1 - x_2$	Tingkat kebebasan sistem dari kesalahan analisis, perancangan, serta implementasi sama pentingnya dengan tingkat kemudahan pengguna untuk belajar serta menggunakan sistem.	x_1 2
$x_1 - x_3$	Tingkat kebebasan sistem dari kesalahan analisis, perancangan, serta implementasi sama pentingnya dengan pemakaian minimal dari sumberdaya sistem	x_1 2
$x_1 - x_4$	Tingkat kebebasan sistem dari kesalahan analisis, perancangan, serta implementasi sama pentingnya dengan Kemampuan sistem untuk menjalankan semua fungsi yang dibutuhkan atau di harapkan	x_4 2
$x_1 - x_5$	Tingkat kebebasan sistem dari kesalahan analisis, perancangan, serta implementasi sama	x_1 2

$x_1 - x_6$	pentingnya dengan tingkat kemampuan sistem/perangkat lunak dalam mencegah akses yang tidak sah	x_1	2	$x_2 - x_3$	kemampuan sistem untuk menjalankan semua fungsi yang dibutuhkan atau di harapkan	x_5	2
$x_1 - x_7$	Tingkat kebebasan sistem dari kesalahan analisis, perancangan, serta implementasi sama pentingnya dengan kemampuan adaptasi sistem	x_1	2	$x_2 - x_6$	Pemakaian minimal dari sumberdaya sistem sama pentingnya dengan tingkat kemampuan sistem/perangkat lunak dalam mencegah akses yang tidak sah	x_2	4
$x_1 - x_8$	Tingkat kebebasan sistem dari kesalahan analisis, perancangan, serta implementasi sama pentingnya dengan tingkat keabsahan sistem/perangkat lunak dari kesalahan-kesalahan (<i>bugs</i>), khususnya ditinjau dari keluaran yang dihasilkan	x_1	3	$x_2 - x_7$	Pemakaian minimal dari sumberdaya sistem sama pentingnya dengan tingkat keabsahan sistem/perangkat lunak dari kesalahan-kesalahan (<i>bugs</i>), khususnya ditinjau dari keluaran yang dihasilkan	x_7	4
$x_2 - x_1$	Tingkat kemudahan pengguna untuk belajar serta menggunakan sistem sama pentingnya dengan pemakaian minimal dari sumberdaya sistem	x_2	2	$x_2 - x_8$	Pemakaian minimal dari sumberdaya sistem sama pentingnya dengan Tingkat kemampuan sistem melanjutkan fungsinya bila terdapat masukan yang tidak sah atau kondisi lingkungan yang menekan	x_8	3
$x_2 - x_4$	Tingkat kemudahan pengguna untuk belajar serta menggunakan sistem sedikit lebih penting dari tingkat kemampuan sistem dalam mencegah akses yang tidak sah	x_4	3	$x_4 - x_1$	Kemampuan sistem untuk menjalankan semua fungsi yang dibutuhkan atau di harapkan sedikit lebih penting dari tingkat kemampuan sistem/perangkat lunak dalam mencegah akses yang tidak sah	x_4	3
$x_2 - x_5$	Tingkat kemudahan pengguna untuk belajar serta menggunakan sistem sedikit lebih penting dari tingkat kemampuan sistem/perangkat lunak dalam mencegah akses yang tidak sah	x_5	3	$x_4 - x_6$	Kemampuan sistem untuk menjalankan semua fungsi yang dibutuhkan atau di harapkan sama pentingnya dengan kemampuan adaptasi sistem	x_6	2
$x_2 - x_6$	Tingkat kemudahan pengguna untuk belajar serta menggunakan sistem sedikit lebih penting dari kemampuan adaptasi sistem	x_2	3	$x_4 - x_7$	Kemampuan sistem untuk menjalankan semua fungsi yang dibutuhkan atau di harapkan sama pentingnya dengan tingkat keabsahan sistem/perangkat lunak dari kesalahan-kesalahan (<i>bugs</i>), khususnya ditinjau dari keluaran yang dihasilkan	x_7	2
$x_2 - x_7$	Pemakaian minimal dari sumberdaya sistem sama pentingnya dengan tingkat keabsahan sistem/perangkat lunak dari kesalahan-kesalahan (<i>bugs</i>), khususnya ditinjau dari keluaran yang dihasilkan	x_7	3	$x_4 - x_8$	Kemampuan sistem untuk menjalankan semua fungsi yang dibutuhkan atau di harapkan sedikit lebih penting dari tingkat kemampuan sistem melanjutkan fungsinya bila terdapat masukan yang tidak sah atau kondisi lingkungan yang menekan	x_8	3
$x_2 - x_8$	Pemakaian minimal dari sumberdaya sistem sama pentingnya dengan tingkat kemampuan sistem melanjutkan fungsinya bila terdapat masukan yang tidak sah atau kondisi lingkungan yang menekan	x_8	3	$x_5 - x_1$	Tingkat kemampuan sistem/perangkat lunak dalam mencegah akses yang tidak sah sedikit lebih penting dari kemampuan adaptasi sitem	x_5	3
$x_2 - x_4$	Pemakaian minimal dari sumberdaya sistem sedikit lebih penting dari	x_4	3	$x_5 - x_7$	Tingkat kemampuan sistem/perangkat lunak dalam mencegah akses yang tidak sah sedikit lebih penting dari tingkat keabsahan sistem/perangkat lunak dari	x_7	3

$x_2 - x_3$	kesalahan-kesalahan (<i>bugs</i>), khususnya ditinjau dari keluaran yang dihasilkan	x_3	3
$x_2 - x_7$	Tingkat kemampuan sistem/perangkat lunak dalam mencegah akses yang tidak sah sedikit lebih penting dari tingkat kemampuan sistem melanjutkan fungsinya bila terdapat masukan yang tidak sah atau kondisi lingkungan yang menekan	x_7	2
$x_2 - x_8$	Kemampuan adaptasi sistem sama pentingnya dengan tingkat keabsahan sistem/perangkat lunak dari kesalahan-kesalahan (<i>bugs</i>), khususnya ditinjau dari keluaran yang dihasilkan	x_8	3
$x_7 - x_8$	Kemampuan adaptasi sistem sedikit lebih penting dari tingkat kemampuan sistem melanjutkan fungsinya bila terdapat masukan yang tidak sah atau kondisi lingkungan yang menekan	x_7	3

Setelah itu dilanjutkan dengan menentukan matriks perbandingan pasangan (*pairwise comparison matrix*) berdasarkan kombinasi karakteristik yang telah diperoleh pada langkah pertama.

TABEL V
Matriks Perbandingan Pasangan

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	1	2	2	2	2	2	2	3
x_2	0.50	1	2	3	3	3	3	3
x_3	0.50	0.50	1	3	2	4	4	3
x_4	0.50	0.33	0.33	1	1	2	2	3
x_5	0.50	0.33	0.50	1.00	1	3	3	3
x_6	0.50	0.33	0.25	0.33	0.33	1	2	3
x_7	0.50	0.33	0.25	0.33	0.33	0.50	1	3
x_8	0.33	0.33	0.33	0.33	0.33	0.33	0.33	1
Σ	4.33	5.17	6.67	11.00	10.00	15.83	17.33	22.00

Tabel V adalah matriks perbandingan pasangan yang digunakan untuk mengumpulkan preferensi relatif antara kriteria-kriteria dari karakteristik luar *software*. Responden memberikan perbandingan dua per dua terhadap kriteria-kriteria tersebut dan menilai sejauh mana satu kriteria lebih penting daripada kriteria lainnya dari karakteristik luar *software*. Selanjutnya, matriks perbandingan pasangan menjadi dasar untuk menghitung nilai-nilai eigen dan vektor eigen.

TABEL VI
Matriks Bobot Nilai Antar Kriteria dan Prioritas

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Prioritas
x_1	0.23	0.39	0.30	0.18	0.20	0.13	0.12	0.14	0.21
x_2	0.12	0.19	0.30	0.27	0.30	0.19	0.17	0.14	0.21
x_3	0.12	0.10	0.15	0.27	0.20	0.25	0.23	0.14	0.18
x_4	0.12	0.06	0.05	0.09	0.10	0.13	0.12	0.14	0.10
x_5	0.12	0.06	0.08	0.09	0.10	0.19	0.17	0.14	0.12
x_6	0.12	0.06	0.04	0.03	0.03	0.06	0.12	0.14	0.07
x_7	0.12	0.06	0.04	0.03	0.03	0.03	0.06	0.14	0.06
x_8	0.08	0.06	0.05	0.03	0.03	0.02	0.02	0.05	0.04

Berdasarkan tabel VI terlihat eigen vektor atau nilai prioritas dari setiap kriteria- kriteria. Selanjutnya menghitung nilai eigen maksimal (λ_{max}) untuk mengetahui nilai rasio konsistensi kriteria dari karakteristik luar *software*.

TABEL VII
Nilai Eigen Maksimal

Prioritas	Jumlah	
0.21	4.33	0.909
0.21	5.17	1.085
0.18	6.67	1.212
0.10	11.00	1.098
0.12	10.00	1.181
0.07	15.83	1.179
0.06	17.33	1.098
0.04	22.00	0.937
λ_{max}		8.7

Selanjutnya menghitung nilai *Consistency Index* (CI) dan *Consistency Ratio* (CR) dimana,

$$CI = \frac{(\lambda_{max} - n)}{(n - 1)} = \frac{8.7 - 8}{(8 - 1)} = 0.1 \tag{3}$$

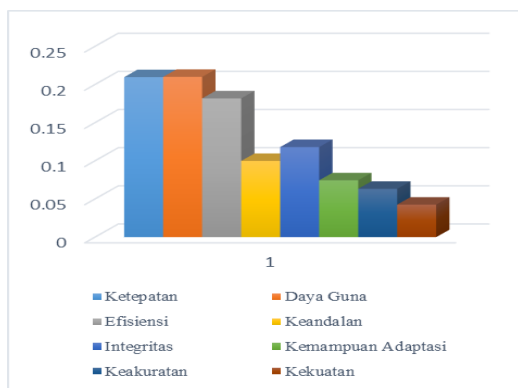
$$CR = \frac{CI}{IR} = \frac{0.1}{1.41} = 0.071 \tag{4}$$

Berdasarkan hasil perhitungan diatas diperoleh bahwa nilai $CR \leq 0.1$ dan menunjukkan hasil perhitungan kriteria konsisten.

IV. PEMBAHASAN

Hasil analisis data menunjukkan nilai *Consistency Ratio* (CR) dari data adalah 0.07 artinya kriteria perbandingan berpasangan yang diberikan oleh pengguna *software* konsisten karena nilai $CR \leq 0.1$ Konsistensi ini dibutuhkan agar perbandingan ini dapat diandalkan dan dapat diolah secara matematis.

Selanjutnya, hasil analisis data yang telah dilakukan dengan metode *Analytic Hierarchy Process* (AHP) menyatakan bahwa *eigen vektor* atau prioritas dari masing-masing kriteria karakteristik luar *software*.



Gambar 2. Grafik Tingkat Prioritas Kriteria Karakteristik Luar *Software*

Eigen vector atau prioritas untuk ketepatan atau tingkat keabsahan sistem dari kesalahan analisis perancangan sebesar 0.21 dan terlihat bahwa pengguna lebih menginginkan *programmer* untuk memprioritaskan ketepatan sistem dari pada kriteria lainnya. Kemudian eigen vector atau prioritas untuk daya guna atau tingkat kemudahan pengguna untuk belajar serta menggunakan sistem sebesar 0.21. Selanjutnya eigen vektor atau prioritas untuk efisiensi atau pemakaian minimal dari sumber daya sistem sebesar 0.18 berada di prioritas ketiga bagi pengguna, selanjutnya eigen vektor atau prioritas efisiensi atau pemakaian minimal dari sumberdaya sistem sebesar 0.18 artinya sebuah *software* seharusnya meminimalisasi penggunaan memori, penggunaan sumberdaya CPU, penggunaan ruang hardisk dan sebagainya sehingga *software* dapat digunakan dalam kondisi/ spesifikasi yang tidak terlalu tinggi, yang dapat dipenuhi oleh kebanyakan calon pengguna. Kriteria interitas mempunyai nilai eigen vektor sebesar 0.12 berada pada prioritas keempat artinya pengguna menginginkan *software* dengan tingkat kemampuan mencegah akses yang tidak sah. Selanjutnya kriteria keandalan berada pada prioritas kelima dengan nilai eogen vektor sebesar 0.1 artinya pengguna menginginkan *software* bisa menjalankan semua fungsi yang dibutuhkan dan diharapkan. Kriteria kemampuan adaptasi atau *software* diharapkan dapat dipakai dalam aplikasi atau lingkungan yang lain dari lingkungan saat *software* diciptakan memiliki nilai eigen vektor sebesar 0.07 yang berada pada urutan prioritas keenam dari delapan kriteria. Kriteria keakuratan dan kekuatan masing-masing berada pada prioritas ketujuh dan kedelapan dari delapan kriteria karakteristik luar.

Selanjutnya, berdasarkan hasil analisis data nilai Consistency Ratio (CR) dari data adalah 0.07 artinya kriteria perbandingan berpasangan yang diberikan oleh pengguna *software* konsisten karena nilai $CR \leq 0,1$. Konsistensi ini dibutuhkan agar perbandingan ini dapat diandalkan dan dapat diolah secara matematis.

V. KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan bahwa karakteristik luar *software* yang menjadi prioritas utama bagi pengguna adalah ketepatan dan daya guna *software*, dengan nilai eigen vektor sebesar 0.21 atau 21%. Hal ini menunjukkan bahwa dalam mendesain *software*, seorang *programmer* dapat menggunakan keputusan ini sebagai panduan utama untuk meningkatkan kualitas *software*,

sehingga dapat lebih baik memenuhi kebutuhan dan harapan pengguna. Prioritas berikutnya dapat ditentukan berdasarkan tingkat prioritas yang telah diidentifikasi dalam penelitian ini.

Hasil penelitian ini memberikan landasan yang kuat bagi *programmer* untuk fokus pada aspek-aspek yang dianggap paling penting oleh pengguna, seperti efisiensi penggunaan sumber daya sistem dan integritas. Dengan memahami tingkat prioritas ini, *programmer* dapat lebih efektif dalam mengalokasikan sumber daya dan upaya dalam meningkatkan kualitas *software*.

Penelitian ini memberikan kontribusi penting dalam konteks meingkatkan kualitas *software*, karena memberikan pandangan yang lebih terperinci mengenai preferensi dan kebutuhan pengguna. Oleh karena itu, implementasi hasil penelitian ini dapat menjadi dasar yang kuat untuk menciptakan *software* yang lebih sesuai dengan ekspektasi dan tuntutan pengguna.

REFERENSI

- [1] A. Nugroho, Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek, Bandung: Informatika Bandung, 2005.
- [2] R. W. Hoyer, B. B. Hoyer, P. B. Crosby and W. E. Deming, What is Quality, Quality Progress, Paul Jonhson, 2001.
- [3] R. Dinita and T. Herlambang, Riset Operasi dan Komputasi, Surabaya: Kresna Insan Bina Prima, 2020.
- [4] D. A. Lestari and E. Nababan, "Implementasi Penerapan Metode Analytical Hierarchy Process (AHP) Dalam Pemilihan Paket Layanan Internet," *BLANTIKA Multidisciplinary Journal*, vol. 1, no. 4, pp. 345-357, 2023.
- [5] J. E. S. Casym and D. N. Oktiaara, "Aplikasi Analytical Hierarchy Process dalam Mengidentifikasi Preferensi Laptop Bagi Mahasiswa," in *Seminar Nasional Teknologi Komputer dan Sains (SAINTEKS)*, Tangerang Selatan, 2020.
- [6] W. Vrayuda and Y. Perwira, "Penerapan Metode Analytical Hierarchy Process (AHP) Dalam Rekomendasi Subsidi Listrik," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 3, no. 3, pp. 202-228, 2021.
- [7] S. Sory, N. B. Ginting and F. Rachmawati, "Sistem Pendukung Keputusan Rekomendasi Game Menggunakan Metode Analytical Hierarchy Process (AHP)," *Jurnal Ekonomi dan Teknik (ETNIK)*, vol. 2, no. 3, p. 218, 2023.
- [8] L. Mayola, M. Afdhal, Rita and M. H. Yuhandri, "Analytical Hierarchy Process (AHP) dalam Seleksi Penerimaan Mahasiswa Baru," *Jurnal KomtekInfo*, vol. 10, no. 2, pp. 81-86, 2023.
- [9] S. Bantun, R. Adawiyah, K. Syaban, D. F. Priambodo, Nirsal, S. P. Lestari and J. Y. Sari, "Implementasi Analytical Hierarchy Process Dan Metode Perbandingan Eksponensial Untuk Pemberian Reward," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 10, no. 1, pp. 252-265, 2023.
- [10] M. Tavana, M. Soltanifar, F. J. Santos and Arteaga, "Analytical Hierarchy Process : Revolution And Evolution," *Springer Nature*, vol. 326, pp. 879-907, 2023.
- [11] R. V. Francozo, L. S. V. U. Junior, E. S. Carrapateira, B. C. S. Pacheco, M. T. Oliveira, G. B. Torsoni and J. Yari, "A Web Based Software For Group Decision With Analytic Hierarchy Process," *MethodsX*, vol. 11, 2023.
- [12] Djaali, Metodologi Penelitian Kuantitatif, Jakarta: PT Bumi Aksara, 2020.
- [13] N. Birrel and M. Ould, A Practical Handbook For Software Development, London: Cambridge University Press, 1998.
- [14] D. Nofriansyah and S. Defit, Multi Criteria Decision Making (MCDM), Yogyakarta: Deepublish, 2017.
- [15] B. Suprpto, Model Sistem dan Penerapan Analytical Hierarchy Process (AHP) Pada Sistem Pendukung Keputusan, Jawa Tengah: CV.ZT CORPORA, 2022.