

Pemilihan Produk *Database* pada Google Cloud Platform Menggunakan Metode *Analytical Hierarchy Process* untuk Aplikasi XYZ

Database Product Selection on Google Cloud Platform Using Analytical Hierarchy Process Method for XYZ Application

Muhammad Abyan Naufal¹, Yusuf Priyandari², Yuniaristanto³

Program Studi Teknik Industri Universitas Sebelas Maret

E-mail: ¹muhammadabyannaufal@student.uns.ac.id, ²priyandari@staff.uns.ac.id,

³yuniaristanto@staff.uns.ac.id

Abstrak

Aplikasi XYZ merupakan aplikasi yang dikembangkan untuk pengguna gawai dengan *operating system* Android. Aplikasi ini berfungsi untuk mengidentifikasi berbagai jenis buah dari foto yang diambil secara langsung maupun dari galeri pengguna. Dalam pengembangannya, aplikasi diharapkan dapat menyimpan informasi dan gambar hasil identifikasi pada penyimpanan *cloud*. Google Cloud Platform, yang ditawarkan oleh Google, merupakan penyedia layanan *cloud computing*. Terdapat setidaknya tiga produk yang ditawarkan oleh Google Cloud Platform untuk mengembangkan *database* aplikasi XYZ, yakni Cloud Firestore, Cloud SQL, dan Cloud Spanner. Tujuan penelitian ini adalah untuk menentukan prioritas pemilihan produk *database*, guna pengembangan aplikasi XYZ dengan metode Analytical Hierarchy Process (AHP). Digunakan empat kriteria, yakni *user needs*, *easy to use*, *security*, dan *cost*. Berdasarkan pengolahan data, *cost* menjadi prioritas utama untuk dipertimbangkan dalam memilih *database*, sementara *disaster recovery* menempati prioritas terakhir. Berdasarkan analisis menggunakan metode AHP, dapat disimpulkan produk *database* yang paling direkomendasikan untuk mengembangkan aplikasi XYZ adalah Cloud Firestore.

Kata kunci: AHP, *Cloud Computing*, *Cloud Database*.

Abstract

The XYZ application is an application developed for smartphone users with the Android operating system. This application serves to identify various types of fruit and plants from photos taken directly or from the user gallery. In its development, the application is expected to be able to store information and images identified on the cloud storage. Google Cloud Platform, offered by Google, is a cloud computing service provider. There are at least three products offered by Google Cloud Platform to develop the XYZ application database, namely Cloud Firestore, Cloud SQL, and Cloud Spanner. The purpose of this study is to determine the priority of selecting database products, for the development of XYZ applications using the Analytical Hierarchy Process (AHP) method. Four criteria are used, namely user needs, easy to use, security, and cost. Based on data processing, cost is the main priority to consider in selecting a database, while disaster recovery takes the last priority. Based on the analysis using the AHP method, it can be concluded that the most recommended database product for developing XYZ applications is Cloud Firestore.

Keywords: AHP, *Cloud Computing*, *Cloud Database*.

1. PENDAHULUAN

Cloud computing atau dalam bahasa Indonesia disebut komputasi awan sudah menjadi istilah umum di era industri 4.0 yang menggaungkan *internet of things*. *Cloud computing* merupakan gabungan antara pemanfaatan teknologi komputer dengan pengembangan berbasis

internet [1]. *Cloud* sendiri adalah metafora dari internet seperti awan yang digambarkan pada diagram jaringan komputer, dan abstraksi dari infrastruktur kompleks yang disembunyikannya [2]. Google sebagai perusahaan multinasional yang bergerak di bidang teknologi memiliki sebuah *platform* yang menyediakan berbagai macam layanan dan produk *cloud computing*, yakni *Google Cloud Platform* (GCP). GCP menawarkan layanan yang memberikan pengembang beberapa produk untuk membangun berbagai program dari situs web sederhana hingga aplikasi terdistribusi yang kompleks di seluruh dunia [3]. Salah satu layanan yang ditawarkan oleh GCP adalah *cloud database*, dengan berbagai varian produk yang memiliki karakteristik berbeda-beda.

Aplikasi XYZ adalah aplikasi berbasis Android yang dikembangkan untuk membantu pengguna dalam mengidentifikasi buah-buahan. Secara teknis, aplikasi XYZ dapat mengidentifikasi berbagai jenis buah dari foto yang diambil melalui aplikasi maupun melalui foto yang tersimpan di galeri, kemudian memberi beberapa informasi terkait buah tersebut, seperti nama umum, famili, genus, ordo, kondisi buah (busuk/segar), dan kandungan nutrisi. Aplikasi XYZ adalah proyek kecil yang diberikan kepada sebuah tim mahasiswa dalam pengawasan pihak Google. Dalam pengembangan aplikasi XYZ, *cloud computing* memiliki peranan sebagai media penyimpanan gambar dan riwayat pengguna yang berisi tanggal identifikasi dan informasi terkait buah tersebut. Media penyimpanan awan ini memberikan solusi ruang penyimpanan yang besar dan dapat diakses dari mana saja dan kapan saja dengan melalui berbagai perangkat yang terhubung dengan internet [4]. Dengan begitu, aplikasi tidak akan menghabiskan memori pengguna untuk menyimpan gambar dan informasi.

Gambar yang merupakan data bertipe objek akan disimpan ke dalam Google Cloud Storage, sementara data riwayat pengguna yang berbentuk teks akan disimpan di *cloud database*. Terdapat tiga produk *cloud database* yang ditawarkan oleh GCP, diantaranya adalah Cloud Firestore, Cloud SQL, dan Cloud Spanner. Cloud Firestore adalah database berbasis dokumen NoSQL yang dibuat untuk penskalaan otomatis, kinerja yang tinggi, dan mengusung kemudahan dalam pengembangan aplikasi [5]. Meskipun antarmuka Firestore memiliki banyak fitur yang sama seperti database tradisional, namun sebagai database NoSQL Firestore berbeda dalam cara menggambarkan hubungan antara objek data. Cloud Firestore juga menawarkan integrasi tanpa batas dengan produk Firebase dan Google Cloud lainnya, termasuk Cloud Functions. Cloud SQL adalah layanan database relasional Google yang dikelola sepenuhnya untuk memudahkan persiapan, pengelolaan, pemeliharaan, dan pengelolaan database relasional di *Google Cloud Platform* [6]. Cloud SQL menyediakan alternatif berbasis cloud untuk database MySQL, PostgreSQL, dan SQL Server. Banyak aplikasi yang berjalan di Compute Engine, App Engine, dan layanan lain di Google Cloud menggunakan Cloud SQL untuk penyimpanan database [7]. Cloud Spanner adalah database unik yang menggabungkan transaksi, kueri SQL, dan struktur relasional dengan skalabilitas database non-relasional atau NoSQL [8]. Cloud Spanner dapat menjangkau beberapa mesin, pusat data, dan wilayah, sehingga distribusi ini dimanfaatkan untuk menangani kumpulan data dan beban kerja yang sangat besar sambil tetap mempertahankan ketersediaan yang sangat tinggi [6].

Tim pengembang perlu memilih produk *cloud database* yang relevan diantara ketiga produk yang ditawarkan GCP. Dalam hal ini, identifikasi dan penyusunan kriteria untuk memilih *cloud database* sangat diperlukan karena dapat memberi pertimbangan bagi tim pengembang untuk mengambil keputusan dalam memilih produk yang tepat. Salah satu metode yang dapat menjawab permasalahan ini adalah penggunaan metode *Analytical Hierarchy Process* (AHP). AHP merupakan suatu metode yang digunakan dalam proses pengambilan keputusan suatu masalah kompleks multikriteria, seperti pemilihan prioritas kebijakan, pemilihan alokasi sumber daya, perencanaan performansi organisasi, dan pemecahan konflik yang lebih objektif dan subjektif [9]. AHP menggunakan hirarki pengambilan keputusan yang berurut mulai dari kriteria, sub kriteria, hingga alternatif [10]. Metode AHP telah banyak digunakan untuk menyelesaikan masalah pengambilan keputusan yang memiliki banyak kriteria, diantaranya adalah untuk memilih sekolah [11], memilih laptop bagi mahasiswa [12], melakukan analisis kinerja gudang [13], hingga memilih supplier [14].

Niemcewicz [15] pada tahun 2021 melakukan penelitian untuk memilih *cloud computing*

provider menggunakan metode AHP, kriteria pemilihan yang digunakan adalah jumlah layanan yang tersedia, jangkauan geografi, kepercayaan terhadap *provider*, jalur akses, dan ketersediaan layanan. Jiang et al. [16] melakukan penelitian untuk mengukur performa layanan SaaS secara objektif dan kuantitatif, sehingga dapat membantu pengguna untuk ketika memilih *cloud service* dengan metode AHP. Kriteria yang digunakan pada penelitian tersebut diantaranya adalah *availability, reliability, security, real-time, easy-to-use, compatibility, dan maintainability*.

Dua penelitian yang telah disebutkan sebelumnya sama-sama menggunakan metode AHP dengan objek penelitian terkait pemilihan *cloud service provider*. Dalam penelitian tersebut peneliti mengidentifikasi kriteria pemilihan *cloud service provider* dan memberikan rekomendasi *cloud service provider* yang terbaik berdasarkan metode AHP. Adapun kontribusi penelitian ini terletak pada objek penelitian yang berbeda, yakni memilih produk *cloud database* dari *cloud service provider* Google Cloud. Kebaruan yang diberikan pada penelitian ini pada kriteria dan subkriteria yang dapat digunakan untuk memilih *cloud database*, yakni dengan menggabungkan kriteria pada penelitian sebelumnya dan membuat sub-sub kriteria sehingga terbentuk hierarki keputusan. Dengan demikian, tujuan penelitian ini adalah (1) menyusun kriteria dan subriteria yang relevan dalam memilih produk *cloud database*, (2) menentukan bobot prioritas, dan (3) menentukan urutan prioritas produk *cloud atabase* dari GCP menggunakan metode AHP. Penelitian ini menghasilkan rekomendasi produk *cloud database* yang relevan bagi pengembangan aplikasi XYZ.

Artikel ini terdiri dari bagian metode penelitian yang memaparkan objek penelitian, pendekatan dan teknik yang digunakan dalam proses pengambilan data, dan penjabaran metode yang digunakan dalam pengolahan dan analisis data guna mendukung pemecahan masalah. Hasil dan pembahasan memaparkan struktur hierarki keputusan, pembobotan prioritas kriteria dan subkriteria, pembobotan alternatif, dan analisis hasil. Bagian terakhir merupakan kesimpulan yang menggambarkan jawaban dari tujuan yang telah ditetapkan sebelumnya.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian dibagi menjadi tiga bagian seperti pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

Tahapan pertama mengidentifikasi alternatif produk *cloud database* dari GCP yang relevan digunakan untuk proyek pengembangan aplikasi XYZ. Berdasarkan studi literatur, ada tiga alternatif produk *database*, yakni Cloud Firestore, Cloud SQL, dan Cloud Spanner. Tahapan kedua adalah menyusun kriteria dan subkriteria yang dapat digunakan untuk memilih atau mendapatkan prioritas rekomendasi *cloud database*. Hasil tahapan ini adalah struktur hierarki sesuai metode AHP yang disampikan pada bagian hasil penelitian. Tahapan ketiga adalah pengambilan data responden untuk penentuan bobot kepentingan antar kriteria dan subkriteria pemilihan. Sebagai kelanjutan tahapan tersebut, dilakukan juga pengambilan data penilaian responden terhadap alternatif produk *cloud database* berdasarkan kriteria dan subkriteria yang telah disusun. Hasil perhitungan menggunakan metode AHP menghasilkan sebuah rekomendasi produk *cloud database* yang dapat digunakan oleh tim pengembang aplikasi.

2.1 Objek dan Sumber Data Penelitian

Objek penelitian adalah pemilihan produk *cloud database* dari *cloud service provider* Google Cloud Platform untuk mengembangkan aplikasi XYZ. *Cloud database* terpilih akan digunakan untuk menyimpan data riwayat pengguna beserta informasinya yang berbentuk teks.

Terdapat tiga alternatif produk *database*, yakni Cloud Firestore, Cloud SQL, dan Cloud Spanner. Ketiganya diputuskan menjadi alternatif berdasarkan studi literatur yang bersumber dari jurnal dan situs resmi GCP.

Sumber data pada penelitian ini adalah data primer dan sekunder. Data primer diperoleh melalui pengisian kuesioner, sementara data sekunder diperoleh dengan melakukan studi literatur. Kuesioner diberikan kepada responden yang dianggap sebagai seorang ahli, yakni pengembang aplikasi yang juga memahami bidang *cloud computing*. Studi literatur dilakukan terhadap buku, jurnal, dokumen standarisasi, dan penelitian sebelumnya yang bertemakan *cloud computing*, *cloud database*, dan metode *Analytical Hierarchy Process (AHP)*.

2.2 Responden

Penelitian ini menggunakan metode *purposive sampling: expert sample* dalam menentukan responden, dimana responden dipilih berdasarkan keahlian dalam keilmuan yang sejalan dengan objek penelitian. *The experts' sampling* dapat diterapkan karena penelitian memiliki kebutuhan untuk memperoleh sudut pandang para ahli tentang masalah dimana pengetahuan para ahli tersebut diperlukan [17]. Responden pada penelitian ini merupakan *cloud architect* yang memahami karakteristik dan penggunaan produk – produk *Google Cloud Platform*.

2.3 Pengolahan Data

Pengolahan data hasil kuesioner menggunakan metode AHP dilakukan dengan langkah-langkah sebagai berikut [18][19]:

1. Mengidentifikasi masalah dan menyusun hierarki yang berisi tujuan, alternatif untuk mencapai tujuan, dan kriteria untuk mengevaluasi alternatif yang ada.
2. Menetapkan prioritas antar hierarki dengan membuat sejumlah penilaian berdasarkan perbandingan berpasangan dari setiap elemen kriteria atau sub kriteria.
3. Menghitung penilaian yang telah didapat untuk menghasilkan prioritas keseluruhan hirarki. Hal-hal yang dilakukan dalam langkah ini adalah:
 - Menjumlahkan nilai-nilai dari setiap kolom pada matriks
 - Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
 - Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata
4. Memeriksa konsistensi dari penilaian. Hal-hal yang dilakukan dalam langkah ini adalah:
 - Menjumlahkan nilai-nilai dari setiap kolom
 - Membagi hasil jumlah setiap baris dengan hasil jumlah setiap kolom untuk mendapatkan nilai eigen atau nilai prioritas masing- masing.
 - Mengalikan nilai eigen dengan jumlah setiap kolom pada matriks pairwise comparison. Kemudian jumlahkan hasil kali tersebut. Hasilnya disebut lamda max (λ_{maks}).
 - Melakukan penghitungan Consistency Index dengan menggunakan rumus:

$$CI = \frac{(\lambda_{maks}-n)}{n-1} \quad (1)$$

dimana:

n = banyaknya elemen.

- Menghitung Consistency Ratio (CR) dengan menggunakan rumus:

$$CR = \frac{CI}{IR} \quad (2)$$

dimana:

CR = Consistency Ratio

CI = Consistency Index

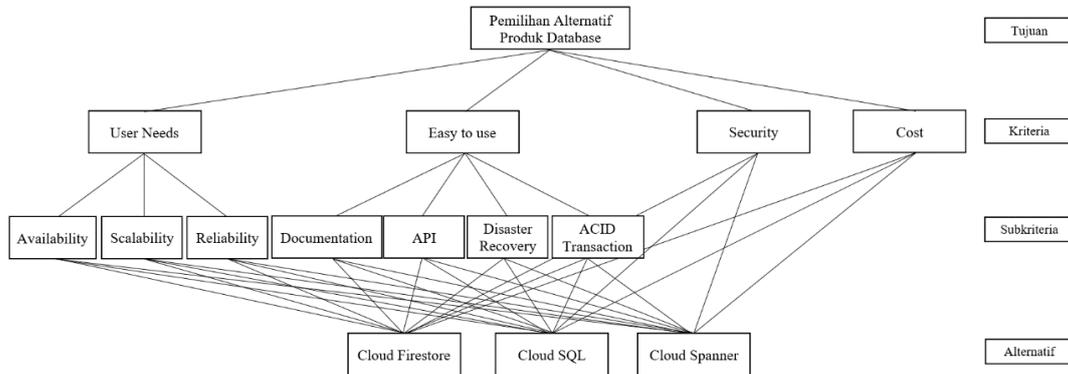
IR = Indeks Random Consistency

- Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10%, maka penilaian data Consistency Ratio (CR) harus diperbaiki

3. HASIL DAN PEMBAHASAN

3.1 Hierarki Keputusan Pemilihan Produk Database

Berdasarkan studi literatur yang telah dilakukan, dipilih 4 kriteria dengan 2 diantaranya memiliki masing-masing 3 dan 4 sub kriteria [16][20]. Gambar 2. Menerangkan kriteria, subkriteria, dan alternatif dalam bentuk struktur hierarki keputusan pemilihan produk *database*.



Gambar 2. Struktur hierarki keputusan pemilihan produk *database*

3.2 Pembobotan Prioritas Kriteria

Setelah hierarki keputusan terbentuk, maka kuesioner dibuat dengan mengacu pada hierarki yang ada untuk kemudian dilakukan pengisian kuesioner oleh *expert*. Hasil kuesioner berupa angka-angka yang melambangkan prioritas dari perbandingan antar kriteria, sub kriteria, hingga antar alternatif. Berdasarkan hasil kuesioner, nilai perbandingan kriteria diolah hingga didapatkan matriks yang dinamakan “*pairwise comparison* kriteria” seperti pada Tabel 1.

Tabel 1. Matriks *Pairwise Comparison* Kriteria

Kriteria	User Needs	Easy to Use	Security	Cost
User Needs	1,000	0,500	0,500	0,167
Easy to Use	2,000	1,000	1,000	0,167
Security	2,000	1,000	1,000	0,333
Cost	6,000	6,000	3,000	1,000
Total	11,000	8,500	5,500	1,667

Berikutnya dilakukan normalisasi matriks dengan membagi nilai masing-masing sel pada Tabel 1. dengan nilai total masing-masing kolom. Hasil normalisasi seperti pada tabel 2.

Tabel 2. Hasil Normalisasi Matriks *Pairwise Comparison* Kriteria

Kriteria	User Needs	Easy to Use	Security	Cost	Total
User Needs	0,091	0,059	0,091	0,100	0,341
Easy to Use	0,182	0,118	0,182	0,100	0,581
Security	0,182	0,118	0,182	0,200	0,681
Cost	0,545	0,706	0,545	0,600	2,397
Total	1,000	1,000	1,000	1,000	4,000

Pembobotan prioritas didapatkan dengan mencari nilai rata-rata setiap baris kriteria atau dengan kata lain membagi nilai setiap sel pada kolom Total di Tabel 2. dengan jumlah kriteria yang digunakan. Didapatkan hasil perhitungan seperti pada Tabel 3.

Tabel 3. Hasil Pembobotan Kriteria terhadap Tujuan

Kriteria	Bobot Kriteria
User Needs	0,085
Easy to Use	0,145
Security	0,170
Cost	0,600

Setelah melakukan pembobotan, maka perlu dilakukan uji konsistensi. Setelah prioritas berhasil ditentukan, maka perlu dilakukan uji konsistensi. Langkah pertama adalah dengan mengalikan matriks *pairwise comparison* Tabel 1. dengan matriks bobot kriteria Tabel 3. sehingga didapatkan matriks seperti pada Tabel 4.

Tabel 4. Calculation of Weighted Sum Kriteria

Kriteria	User Needs	Easy to Use	Security	Cost	Weighted Sum
User Needs	0,085	0,073	0,085	0,100	0,343
Easy to Use	0,170	0,145	0,170	0,100	0,586
Security	0,170	0,145	0,170	0,200	0,686
Cost	0,511	0,872	0,511	0,599	2,493

Langkah berikutnya adalah mencari nilai Lambda maksimum (λ) dengan terlebih dahulu mencari nilai Lambda (λ) tiap kriteria. Nilai Lambda (λ) tiap kriteria didapat dari pembagian nilai *Weighted Sum* pada Tabel 4. dengan nilai pembobotan pada Tabel 3. Lambda maksimum (λ) dihitung dengan mencari rata-rata nilai Lambda (λ) tiap kriteria seperti pada Tabel 5.

Tabel 5. Perhitungan Lamda (λ) Kriteria

Kriteria	Lambda (λ)
User Needs	4,026
Easy to Use	4,031
Security	4,026
Cost	4,161
Total	16,244
Lambda maks (λ)	4,061

Dengan menggunakan data yang ada, diperoleh nilai $CI = \frac{(4,061-4)}{(4-1)} = 0,0203$ sehingga nilai $CR = \frac{0,0203}{0,9} = 0,023$. Nilai $CR < 0,1$, maka dapat disimpulkan bahwa pendapat/persepsi yang diberikan oleh *expert* adalah konsisten.

3.3 Pembobotan Prioritas Sub Kriteria User Needs dan Easy to Use

Sama halnya dengan kriteria, pengolahan data pembobotan sub kriteria melalui tahapan pembuatan matriks *pairwise comparison*, normalisasi matriks *pairwise comparison*, pembobotan sub kriteria, *Calculation of Weighted Sum*, perhitungan lamda, dan perhitungan nilai CR. Tabel 6. hingga Tabel 10. merupakan pengolahan data sub kriteria *user needs*.

Tabel 6. Matriks *Pairwise Comparison* Sub Kriteria *User Needs*

Sub Kriteria	Availability	Scalability	Reliability
Availability	1,000	7,000	2,000
Scalability	0,143	1,000	0,200
Reliability	0,500	5,000	1,000
Total	11,000	8,500	5,500

Tabel 7. Normalisasi Matriks *Pairwise Comparison* Sub Kriteria *User Needs*

Sub Kriteria	Availability	Scalability	Reliability	Total
Availability	0,609	0,538	0,625	1,772
Scalability	0,087	0,077	0,063	0,226
Reliability	0,304	0,385	0,313	1,001
Total	1,000	1,000	1,000	3,000

Pembobotan prioritas didapatkan dengan mencari nilai rata-rata setiap baris subkriteria atau dengan kata lain membagi nilai setiap sel pada kolom total di Tabel 7. dengan jumlah subkriteria yang digunakan, yakni 3. Pembobotan tersebut merupakan pembobotan subkriteria terhadap kriteria, sehingga perlu dilakukan juga pembobotan terhadap tujuan guna perhitungan pada bagian pembobotan alternatif. Pembobotan terhadap tujuan dilakukan dengan mengalikan hasil pembobotan terhadap kriteria dengan nilai pembobotan kriteria, dalam hal ini adalah pembobotan kriteria *User Needs* yakni sebesar 0.085. Didapatkan hasil perhitungan seperti pada Tabel 8.

Tabel 8. Hasil Pembobotan Sub Kriteria *User Needs*

Sub Kriteria	Bobot terhadap Kriteria	Bobot terhadap Tujuan
Availability	0,591	0,050
Scalability	0,075	0,007
Reliability	0,334	0,028

Sub kriteria *Availability* adalah sub kriteria dengan nilai pembobotan tertinggi dibanding kriteria lainnya, disusul dengan sub kriteria *Reliability* lalu *Scalability*. Tahap uji konsistensi juga perlu dilakukan, langkah pertama adalah dengan mengalikan matriks *pairwise comparison* dengan matriks bobot subkriteria sehingga didapatkan matriks seperti pada Tabel 9.

Tabel 9. Calculation of Weighted Sum Subkriteria *User Needs*

Subkriteria	Availability	Scalability	Reliability	Weighted Sum
Availability	0,591	0,528	0,668	1,787
Scalability	0,084	0,075	0,067	0,227
Reliability	0,295	0,377	0,334	1,006

Nilai Lambda (λ) tiap kriteria didapat dari pembagian nilai *Weighted Sum* pada Tabel 9 dengan nilai pembobotan pada Tabel 8. Lambda maksimum (λ) dihitung dengan mencari rata-rata nilai Lambda (λ) tiap kriteria seperti pada Tabel 10.

Tabel 10. Perhitungan Lambda (λ) Subkriteria *User Needs*

Kriteria	Lambda (λ)
Availability	3,024
Scalability	3,003
Reliability	3,015
Total	9,043
Lambda maks (λ)	3,014

Dengan menggunakan data yang ada, diperoleh nilai $CI = \frac{(3,014-3)}{(3-1)} = 0,007$ sehingga nilai $CR = \frac{0,007}{0,58} = 0,012$. Nilai $CR < 0,1$, maka dapat disimpulkan bahwa pendapat/persepsi yang diberikan oleh *expert* adalah konsisten. Berikutnya, Tabel 11. hingga Tabel 15. merupakan pengolahan data sub kriteria *easy to use*.

Tabel 11. Matriks *Pairwise Comparison* Sub Kriteria *Easy to Use*

Sub Kriteria	Documentation	API	Disaster Recovery	ACID Transactions
Documentation	1,000	3,000	9,000	3,000
API	0,333	1,000	5,000	1,000
Disaster Recovery	0,111	0,200	1,000	0,143
ACID Transaction	0,333	1,000	7,000	1,000
Total	1,778	5,200	22,000	5,143

Tabel 12. Normalisasi Matriks *Pairwise Comparison* Sub Kriteria *Easy to Use*

Sub Kriteria	Documentation	API	Disaster Recovery	ACID Transactions	Total
Documentation	0,563	0,577	0,409	0,583	2,132
API	0,188	0,192	0,227	0,194	0,802
Disaster Recovery	0,063	0,038	0,045	0,028	0,174
ACID Transaction	0,188	0,192	0,318	0,194	0,892
Total	1,000	1,000	1,000	1,000	4,000

Pembobotan prioritas didapatkan dengan mencari nilai rata-rata setiap baris subkriteria atau dengan kata lain membagi nilai setiap sel pada kolom Total di Tabel 12. dengan jumlah subkriteria yang digunakan, yakni 4. Pembobotan tersebut merupakan pembobotan subkriteria terhadap kriteria, sehingga perlu dilakukan juga pembobotan terhadap tujuan guna perhitungan pada bagian pembobotan alternatif. Pembobotan terhadap tujuan dilakukan dengan mengalikan hasil pembobotan terhadap kriteria dengan nilai pembobotan kriteria, dalam hal ini adalah pembobotan kriteria *Easy to Use* yakni sebesar 0.145. Didapatkan hasil perhitungan seperti pada Tabel 8.

Tabel 13. Hasil Pembobotan Sub Kriteria *Easy to Use*

Sub Kriteria	Bobot terhadap Kriteria	Bobot terhadap Tujuan
Documentation	0,533	0,077
API	0,200	0,029
Disaster Recovery	0,044	0,006
ACID Transaction	0,223	0,032

Sub kriteria *Documentation* adalah sub kriteria dengan nilai pembobotan tertinggi dibanding kriteria lainnya, disusul dengan sub kriteria *ACID Transaction*, *API*, lalu *Disaster Recovery*. Tahap uji konsistensi juga perlu dilakukan, langkah pertama adalah dengan mengalikan matriks *pairwise comparison* dengan matriks bobot subkriteria sehingga didapatkan matriks seperti pada Tabel 14.

Tabel 14. Calculation of Weighted Sum Subkriteria *Easy to Use*

Subkriteria	Docum.	API	Disaster Rec.	ACID Trans.	Weighted Sum
Docum.	0,533	0,601	0,392	0,669	2,195
API	0,178	0,200	0,218	0,223	0,819
Disaster Rec.	0,059	0,040	0,044	0,032	0,175
ACID Trans.	0,178	0,200	0,305	0,223	0,906

Nilai Lambda (λ) tiap kriteria didapat dari pembagian nilai *Weighted Sum* pada Tabel 14 dengan nilai pembobotan pada Tabel 13. Lambda maksimum (λ) dihitung dengan mencari rata-rata nilai Lambda (λ) tiap kriteria seperti pada Tabel 15.

Tabel 15. Perhitungan Lamda (λ) Kriteria

Kriteria	Lambda (λ)
Docum.	4,119
API	4,087
Disaster Rec.	4,012
ACID Trans.	4,061
Total	16,279
Lambda maks (λ)	4,070

Dengan menggunakan data yang ada, diperoleh nilai $CI = \frac{(4,070-4)}{(4-1)} = 0,023$ sehingga nilai $CR = \frac{0,023}{0,9} = 0,026$. Nilai $CR < 0.1$, maka dapat disimpulkan bahwa pendapat/persepsi yang diberikan oleh *expert* adalah konsisten.

3.4 Pembobotan Prioritas Alternatif

Pada bagian ini dilakukan pembobotan alternatif produk *database* terhadap kriteria dan subkriteria yang ada. Kriteria dan subkriteria yang digunakan dalam pembobotan alternatif adalah *Availability*, *Scalability*, *Reliability*, *Documentation*, *API*, *Disaster Recovery*, *ACID Transaction*, *Cost*, dan *Security*. Tahapan pembobotan prioritas alternatif seperti pembobotan yang lainnya, hanya saja tidak perlu dilakukan uji konsistensi sehingga tahapannya adalah pembuatan matriks *pairwise comparison*, normalisasi matriks *pairwise comparison*, dan pembobotan.

Ketiga alternatif yakni *Cloud SQL*, *Cloud Spanner*, dan *Cloud Firestore* dibandingkan dengan kriteria dan subkriteria yang disebutkan sebelumnya sehingga terdapat banyak tahap pengolahan data pada bagian ini. Maka dari itu, Tabel 16. merangkum hasil pembobotan alternatif

terhadap keseluruhan kriteria dan subkriteria.

Tabel 16. Matriks Bobot Alternatif terhadap Keseluruhan Kriteria

Alternatif	Availability	Scalability	Reliability	Documentation	API	Disaster Recovery	ACID Transaction	Security	Cost
Cloud SQL	0,106	0,083	0,633	0,072	0,074	0,669	0,091	0,260	0,079
Cloud Spanner	0,260	0,193	0,260	0,697	0,643	0,267	0,455	0,106	0,153
Cloud Firestore	0,633	0,724	0,106	0,232	0,283	0,064	0,455	0,633	0,768

Matriks pembobotan alternatif terhadap kriteria dan subkriteria dikalikan dengan matriks pembobotan kriteria dan subkriteria terhadap tujuan yang telah dihitung sebelumnya (Tabel 3, Tabel 8, Tabel 13.), sehingga dihasilkan matriks pembobotan prioritas alternatif produk *database* seperti pada Tabel 17.

Tabel 17. Matriks Pembobotan Alternatif Produk Database

Alternatif	Bobot
Cloud SQL	0,130
Cloud Spanner	0,221
Cloud Firestore	0,649

3.5 Pembahasan

Berdasarkan pengolahan data yang sudah dilakukan, diketahui bahwa semua nilai CR pada pembobotan kriteria dan subkriteria di bawah 0,1. Hal ini berarti tidak terjadinya inkonsistensi pendapat yang diberikan responden, sehingga tidak diperlukannya pengambilan ulang data. Prioritas kriteria yang dipertimbangkan dalam memilih alternatif produk *cloud database* diketahui berdasarkan nilai pembobotan. Prioritas diurutkan berdasar nilai pembobotan terbesar hingga terkecil. Tabel 18. menampilkan urutan kriteria yang perlu diperhatikan dalam pemilihan alternatif produk *cloud database*.

Tabel 18. Prioritas Kriteria untuk Pemilihan Database

Kriteria	Bobot Kriteria	Prioritas
User Needs	0,085	4
Easy to Use	0,145	3
Security	0,170	2
Cost	0,600	1

Berdasarkan Tabel 18, kriteria *Cost* memiliki nilai bobot kriteria terbesar, sehingga menjadi kriteria dengan prioritas utama untuk dipertimbangkan ketika memilih produk *cloud database*. Kriteria yang perlu dipertimbangkan selanjutnya adalah *security*, kemudian *easy to use*, dan terakhir adalah *user needs*. Kendati demikian, kriteria *user needs* dan *easy to use* memiliki subkriteria, sehingga perlu diperhitungkan pula pembobotan dan prioritas dalam memilih alternatif produk *cloud database*. Prioritas sub kriteria *user needs* dan *easy to use* adalah seperti pada Tabel 19 dan Tabel 20.

Tabel 19. Prioritas Subkriteria User Needs untuk Pemilihan Database

Subkriteria	Bobot terhadap Kriteria	Bobot terhadap Tujuan	Prioritas
Availability	0,591	0,050	1
Scalability	0,075	0,007	3
Reliability	0,334	0,028	2

Tabel 20. Prioritas Subkriteria Easy to Use untuk Pemilihan Database

Subkriteria	Bobot terhadap Kriteria	Bobot terhadap Tujuan	Prioritas
Documentation	0,533	0,077	1
API	0,200	0,029	3
Disaster Recovery	0,044	0,006	4
ACID Transaction	0,223	0,032	2

Terdapat dua kolom pembobotan pada Tabel 19 dan 20. Kolom pembobotan pertama adalah bobot subkriteria terhadap masing-masing kriteria (contoh: *availability* terhadap *user needs*). Kolom pembobotan kedua adalah bobot subkriteria terhadap tujuan, yakni pemilihan alternatif produk *cloud database*. Perhitungan pembobotan subkriteria terhadap tujuan dilakukan dengan mengalikan nilai bobot terhadap kriteria dengan nilai bobot kriteria masing-masing; *user needs* dan *easy to use*. Hal ini perlu dilakukan mengingat kriteria *security* dan *cost* tidak memiliki sub kriteria, sehingga nilai pembobotan keduanya akan digunakan langsung dalam perhitungan pembobotan prioritas alternatif, sementara kriteria *user needs* dan *easy to use* akan diwakilkan oleh subkriteria masing-masing.

Tabel 21. menunjukkan urutan prioritas kriteria dan sub kriteria yang digunakan dalam memilih alternatif produk *database*.

Tabel 21. Prioritas Kriteria dan Sub Kriteria untuk Pemilihan Database

Kriteria/Subkriteria	Bobot	Prioritas
Availability	0,050	4
Scalability	0,007	8
Reliability	0,028	7
Documentation	0,077	3
API	0,029	6
Disaster Recovery	0,006	9
ACID Transaction	0,032	5
Security	0,170	2
Cost	0,600	1

Berdasarkan Tabel 21, maka prioritas kriteria yang perlu diperhatikan secara berurutan dalam memilih produk *cloud database* adalah *cost*, *security*, *documentation*, *availability*, *ACID transaction*, *API*, *reliability*, *scalability*, dan *disaster recovery*.

Setelah menghitung bobot kriteria dan sub kriteria, selanjutnya dilakukan pemilihan alternatif produk *database*. Setiap kriteria dan sub kriteria dibandingkan terhadap alternatif yang ada dan dilakukan pembobotan. Sehingga diperoleh hasil akhir pembobotan dan urutan prioritas alternatif produk *database* seperti pada Tabel 22.

Tabel 22. Matriks Prioritas Alternatif Produk Database

Alternatif	Bobot	Prioritas
Cloud SQL	0,130	3
Cloud Spanner	0,221	2
Cloud Firestore	0,649	1

Berdasarkan hasil pada Tabel 20, dapat diketahui produk *database* yang diprioritaskan untuk dipilih adalah Cloud Firestore, karena memiliki nilai bobot terbesar dibandingkan dua produk lainnya. Implikasi manajerial yang dapat diterapkan pada pengembangan aplikasi XYZ adalah memperhatikan *user needs*, *easy to use*, *cost*, *security*, *availability*, *reliability*, *scalability*, *documentation*, *API*, *disaster recovery*, dan *ACID transaction* ketika akan memilih produk *cloud database*. Dari semua kriteria yang perlu diperhatikan, *cost* adalah prioritas utama untuk dipertimbangkan sementara *disaster recovery* merupakan prioritas terakhir. Pihak pengembang juga dapat mempertimbangkan Cloud Firestore sebagai pilihan pertama ketika akan memilih *cloud database* dari Google Cloud Platform. Terakhir, pengembang dapat membangun *micorservice* menggunakan arsitektur dan *cloud service* yang telah dirancang untuk mengembangkan aplikasi XYZ.

4. KESIMPULAN DAN SARAN

Kesimpulan yang diperoleh dari penelitian ini adalah, kriteria yang digunakan dalam pemilihan alternatif produk *database* untuk pengembangan aplikasi XYZ adalah *user needs*, *easy to use*, *security*, dan *cost*. Kriteria *user needs* memiliki sub kriteria, yakni *availability*, *scalability*, dan *reliability*. Kriteria *easy to use* memiliki sub kriteria *documentation*, *API*, *disaster recovery*, dan *ACID transaction*. Sementara untuk kriteria *security* dan *cost* tidak memiliki sub kriteria.

Hal yang menjadi prioritas utama dalam memilih produk *database* adalah *cost* (0,600), sementara hal terakhir yang menjadi pertimbangan dalam memilih produk *database* adalah *disaster recovery* (0,006). Produk *database* yang sangat disarankan untuk mengembangkan aplikasi XYZ adalah Cloud Firestore (0,649). Saran untuk pengembangan penelitian ini adalah mengembangkan kriteria dan subkriteria untuk memilih *cloud database* pada metode *Analytic Hierarchy Process* (AHP), terlebih pada kriteria *security* dan *cost*.

DAFTAR PUSTAKA

- [1] I. Agus, F. Destiawati, and H. Dhika, "Perbandingan Cloud Computing Microsoft Onedrive, Dropbox, dan Google Drive," *Fakt. Exacta*, vol. 12, no. 1, p. 20, 2019, doi: 10.30998/faktorexacta.v12i1.3631.
- [2] H. Dhika, T. Akhirina, D. Mustari, and F. Destiawati, "Pemanfaatan Teknologi Cloud Computing sebagai Media Penyimpanan Data," *J. PkM Pengabd. Kpd. Masy.*, vol. 2, no. 03, p. 221, 2019, doi: 10.30998/jurnalpkm.v2i03.3144.
- [3] S. Challita, F. Zalila, C. Gourdin, and P. Merle, "A precise model for Google cloud platform," *Proc. - 2018 IEEE Int. Conf. Cloud Eng. IC2E 2018*, pp. 177–183, 2018, doi: 10.1109/IC2E.2018.00041.
- [4] R. J. Gunadi, R. Tanone, and Y. R. Beeh, "Penerapan Firebase Cloud Storage Pada Aplikasi Mobile Android Untuk Melakukan Penyimpanan Image Lahan Pertanian," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 282–291, 2020.
- [5] Google, "Firestore documentation." <https://cloud.google.com/firestore/docs> (accessed Jun. 24, 2022).
- [6] N. Sabharwal and S. G. Edward, *Hands On Google Cloud SQL and Cloud Spanner*. 2020. doi: 10.1007/978-1-4842-5537-7.
- [7] Google, "Cloud SQL documentation." <https://cloud.google.com/sql/docs> (accessed Jun. 24, 2022).
- [8] P. Vergadia, "What is Cloud Spanner." <https://cloud.google.com/blog/topics/developers-practitioners/what-cloud-spanner> (accessed Jun. 24, 2022).
- [9] Masitha, D. Hartama, and A. Wanto, "Analisa Metode (AHP) Pada Pembelian Sepatu Sekolah Berdasarkan Konsumen," *Semin. Nas. Sains dan Teknol. Inf.*, pp. 338–342, 2018, [Online]. Available: <http://seminar-id.com/seminas-sensasi2018.htmlPage%7C338>
- [10] R. Aurachman, "Proses Pengambilan Data Pada AHP (Analytical Hierarchy Process) Menggunakan Prinsip Closed Loop Control System," *J. Integr. Sist. Ind.*, vol. 6, no. 1, pp. 55–64, 2019.
- [11] N.- Narti, S. Sriyadi, N. Rahmayani, and M. Syarif, "Pengambilan Keputusan Memilih Sekolah Dengan Metode AHP," *J. Inform.*, vol. 6, no. 1, pp. 143–150, 2019, doi: 10.31311/ji.v6i1.5552.
- [12] J. E. S. Casym and D. N. Oktiara, "Aplikasi Analytical Hierarchy Process dalam Mengidentifikasi Preferensi Laptop Bagi Mahasiswa," *Semin. Nas. Teknol. Komput. Sains*, pp. 636–640, 2020.
- [13] Q. A. Jalal and W. Safitri, "Analisis Kinerja Gudang Dengan Pendekatan Key Performance Indicator (Kpi) Dan Analytical Hierarchy Process (Ahp)," *J. Ilm. Tek. Ind.*, vol. 6, no. 2, pp. 71–78, 2019, doi: 10.24912/jitiuntar.v6i2.4086.
- [14] W. Pratama, Y. Sianturi, A. Silaen, I. Sitorus, and B. Siboro, "PEMILIHAN SUPPLIER

- LAMPU UV DALAM PEMBUATAN BOX STERILIZATION MENGGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS,” *J. Ilm. Tek. Ind.*, vol. 9, no. 2, pp. 158–166, 2021.
- [15] P. Niemcewicz, “The use of the multi-criteria AHP method to select a cloud computing provider,” *Procedia Comput. Sci.*, vol. 192, pp. 2558–2567, 2021, doi: 10.1016/j.procs.2021.09.025.
- [16] T. Gao, T. Li, R. Jiang, R. Duan, R. Zhu, and M. Yang, “A Research About Trustworthiness Metric Method of SaaS Services Based on AHP,” *ICCCS 2018 Cloud Comput. Secur.*, vol. 11063, pp. 207–218, 2018, doi: 10.1007/978-3-030-00006-6.
- [17] N. Apostolopoulos and P. Liargovas, “Regional parameters and solar energy enterprises: Purposive sampling and group AHP approach,” *Int. J. Energy Sect. Manag.*, vol. 10, no. 1, pp. 19–37, 2016, doi: 10.1108/IJESM-11-2014-0009.
- [18] T. L. Saaty, “Decision making with the Analytic Hierarchy Process,” *Int. J. Serv. Sci.*, vol. 1, no. 1, pp. 83–98, 2008, doi: 10.1504/ijssci.2008.017590.
- [19] E. Mu and M. Pereyra-Rojas, “Understanding the Analytic Hierarchy Process,” in *Practical Decision Making*, 2017, pp. 7–22. doi: 10.1007/978-3-319-33861-3_2.
- [20] ISO/IEC9126, “International Standard ISO / IEC 9126 Software Engineering — Product quality ISO 9126 - Content,” *Int. Stand. ISO/IEC 9126, Softw. Eng. Qual.*, pp. 1–8, 2001.