**BAB 2**

**KAJIAN PUSTAKA**

* 1. **Perkembangan *Transportation Service Application***

Berbagai manfaat yang dirasakan oleh masyarakat Jakarta tentang *transportation service application* seperti biaya transportasi cenderung lebih murah, pemesanan jasa transportasi yang mudah, efisien dan cepat, menjadikan *transportation service application* ini mengalami kemajuan yang sangat pesat. *Transportation service application* merupakan sebuah aplikasi yang menyediakan layanan berbagi kendaraan pada suatu waktu dalam waktu singkat [3 Andrew].

*Transportation service application* dapat di unduh secara gratis melalui smartphone di Google Play untuk Android operating system atau di App Store untuk pengguna iPhone. *Transportation service application* yang memiliki jumlah unduhan tertinggi adalah GO-JEK dan GRAB. Kedua aplikasi ini tidak hanya melayani transportasi seperti “ojek” atau “taxi”, melainkan sangat memanjakan penggunanya dengan fitur-fitur yang dapat memberikan banyak kemudahan untuk para penggunanya seperti pengiriman barang, *delivery* makanan, dan lain sebagainya.

**2.1.1 GO-JEK**

GO-JEK adalah sebuah perusahaan yang didirikan oleh Nadiem Makarim pada juni 2010. GO-JEK merupakan perusahaan teknologi berjiwa sosial yang bertujuan untuk meningkatkan kesejahteraan pekerja di berbagai sektor informal di Indonesia. Kami bermitra dengan sekitar 200.000 pengendara ojek yang berpengalaman dan terpercaya di Indonesia, untuk menyediakan berbagai macam layanan, termasuk transportasi dan pesan antar makanan. GO-JEK telah resmi beroperasi di 10 kota besar di Indonesia, termasuk Jakarta, Bandung, Bali, Surabaya, Makassar, Yogyakarta, Medan, Semarang, Palembang, dan Balikpapan dengan rencana pengembangan di kota-kota lainnya pada tahun mendatang.

Kegiatan GO-JEK bertumpu pada tiga nilai pokok: kecepatan, inovasi, dan dampak sosial. Para *driver* GO-JEK mengatakan bahwa pendapatan mereka meningkat semenjak bergabung sebagai mitra, mereka juga mendapatkan santunan kesehatan dan kecelakaan, serta mendapat akses ke lebih banyak pelanggan melalui aplikasi GO-JEK [GOJEKweb]. Logo GO-JEK dapat dilihat pada Figure ?



Figure ? Logo GO-JEK

GO-JEK memiliki beberapa layanan yang akan di jabarkan pada Tabel ? [GOJEKweb].

Tabel ? Layanan GO-JEK [GOJEKweb]

|  |  |  |
| --- | --- | --- |
| **Layanan** | **Gambar** | **Deskripsi** |
| GO-RIDE | GORIDE | GO-RIDE adalah layanan transportasi sepeda motor yang dapat mengantar pengguna GO-JEK ke berbagai tempat, lebih mudah dan lebih cepat. |
| GO-CAR | GOCAR | GO-CAR adalah layanan transportasi menggunakan mobil untuk mengantar pengguna GO-JEK kemanapun dengan nyaman. |
| GO-FOOD | GOFOOD | GO-FOOD adalah layanan pesan antar makanan nomor 1 di Indonesia. GO-JEK memiliki lebih dari 30.000 daftar restoran. |
| GO-SEND | GOSEND | GO-SEND adalah layanan kurir instan yang dapat digunakan untuk mengirim surat dan barang dalam waktu 60 menit. |
| GO-MART | GOMART | GO-MART adalah layanan yang bisa digunakan untuk berbelanja ribuan jenis barang dari berbagai macam toko. |
| GO-BOX | GOBOX | GO-BOX adalah layanan pindah barang ukuran besar menggunakan truk bak/blind van. |
| GO-MASSAGE | GOMAS | GO-MASSAGE adalah layanan jasa pijat kesehatan profesional yang datang ke rumah pengguna GO-JEK. |
| GO-CLEAN | GOCLEAN | GO-CLEAN adalah layanan jasa kebersihan profesional untuk membersihkan kamar kos, rumah dan kantor pengguna GO-JEK. |
| GO-GLAM | GOGLAM | GO-GLAM adalah layanan jasa perawatan kecantikan untuk manicure-pedicure, cream bath, waxing, dan lainnya yang datang ke rumah pengguna GO-JEK. |
| GO-TIX | GOTIX | GO-TIX adalah layanan informasi acara dengan akses pembelian dan pengantaran tiket langsung ke tangan pengguna GO-JEK. |
| GO-BUSWAY | GOBUS | GO-BUSWAY adalah layanan untuk memonitor jadwal layanan bus TransJakarta dan memesan GO-RIDE untuk mengantar pengguna GO-JEK ke halte yang dimaksud. |
| GO-PAY | GOPAY | GO-PAY adalah layanan dompet virtual untuk transaksi pengguna GO-JEK di dalam aplikasi GO-JEK yang bias diisi ulang melalui ATM, M-Banking atau Internet Banking BCA, BRI dan Mandiri. |

Pengukuran kualitas aplikasi GO-JEK akan menggunakan versi yang di update terakhir pada 9 mei 2016. Pada versi ini, GO-JEK *home screen* memiliki *user interface* yang mudah untuk mengakses semua layanan GO-JEK dan untuk layanan GO-RIDE, GO-FOOD dan GO-MART mengalami pembaharuan tampilan. GO-JEK telah menyediakan fitur GO-PAY, sehingga pengguna dapat dengan mudah melakukan pembayaran non tunai untuk semua layanan GO-JEK dengan melakukan *top up* di ATM, M-Banking atau Internet Banking BCA, BRI dan Mandiri. Selain itu GO-JEK memberikan layanan baru yaitu GO-CAR dimana pengguna dapat memesan mobil pribadi yang sama fungsinya dengan GO-RIDE [Google play gojek].

**2.1.2 GRAB**

GRAB merupakan aplikasi layanan transportasi terpopuler di Asia Tenggara yang menyediakan layanan transportasi untuk menghubungkan lebih dari 10 juta penumpang dan 185.000 pengemudi di seluruh wilayah Asia Tenggara Aplikasi GRAB menawarkan 5 pilihan layanan transportasi mulai dari taksi, mobil pribadi, sepeda motor hingga pengiriman paket untuk memenuhi kebutuhan penumpang di Singapura, Malaysia, Thailand, Filipina, Vietnam dan Indonesia. [GRAB].

GRAB didirikan oleh Anthony Tan dan Hooi Ling Tan yang merupakan warga negara Malaysia, mereka melihat adanya dampak negatif dari tidak efisiennya sistem transportasi yang ada pada saat itu. Merekapun memiliki ide untuk membuat aplikasi pemesanan transportasi, khususnya taksi, yang kemudian menobatkan mereka sebagai finalis dalam Kontes Harvard Business School’s 2011 Business Plan [GRAB]. Logo GRAB dapat dilihat pada Figure ?



Figure ? Logo GO-JEK

GRAB memiliki beberapa layanan yang akan di jabarkan pada Tabel ? [GRAB].

Tabel ? Layanan GRAB [GRAB]

|  |  |  |
| --- | --- | --- |
| **Layanan** | **Gambar** | **Deskripsi** |
| GRABTaxi | taxi | GRABTaxi adalah layanan yang membantu penumpang menemukan pengemudi taksi terdekat pada aplikasi GRAB. Jaringan armada taksi terluas yang menawarkan pengalaman berkendara yang paling efisien dan aman dengan tarif sesuai argo dan pengemudi taksi yang berlisensi untuk memastikan layanan berkendara yang aman dan nyaman. |
| GRABCar | car | GRABCar adalah layanan penyewaan kendaraan pribadi dengan supir yang menghadirkan kebebasan pilihan berkendara yang nyaman dan gaya dengan menyediakan asuransi kecelakaan bagi maksimal enam orang dalam satu kendaraan. |
| GRABBike | bike | GRABBike adalah layanan transportasi menggunakan sepeda motor yang cepat melintasi kemacetan Jakarta. |
| GRABExpress | express | GRABExpress adalah layanan kurir ekspres berbasis aplikasi yang menjanjikan Kecepatan, Kepastian, dan yang paling utama adalah Keamanan. |
| GRABFood | food | GRABFood adalah layanan pemesanan makanan dengan menjelajahi dunia gastronomi melalui beragam restoran pilihan berkualitas dan diantarkan ke tempat tujuan pengguna dengan kualitas yang tetap terjaga. |
| GRABPay | pay | GRABPay adalah layanan dompet virtual untuk transaksi pengguna GRAB dengan mendaftarkan kartu debit atau kredit |

Pengukuran kualitas aplikasi GRAB akan menggunakan versi yang di update terakhir pada 14 mei 2016. Pada versi ini, GRAB menambahkan layanan baru yaitu GRABFood versi beta. Versi terbaru ini juga termasuk *bug fixes and code enhancements* untuk meningkatkan kestabilan aplikasi GRAB [Google play GRAB].

* 1. ***Software Quality Assurance (SQA)***

Menurut IEEE *Glossary*, *Software Quality Assurance* (SQA) merupakan pola terencana dan sistematis yang diperlukan untuk memastikan item atau produk sesuai dengan kebutuhan teknis yang telah ditetapkan. *Software Quality Assurance* (SQA) juga menjadi serangkaian kegiatan evaluasi yang dirancang untuk mengembangkan atau memproduksi suatu produk. SQA didasarkan pada perencanaan dan penerapan dari berbagai tindakan yang diintegrasikan ke dalam semua tahapan proses pengembangan perangkat lunak. Hal ini dilakukan untuk mendukung kepercayaan pengguna bahwa produk perangkat lunak akan memenuhi semua kebutuhan teknis. Meskipun perencanaan dan pelaksanaan yang dilakukan secara sistematis, lingkup SQA tidak termasuk maintenance, jadwal, dan masalah anggaran. Definisi SQA diperluas sesuai dengan konsep dasar standar-standar dari model kualitas yang ada. [Galin]

Kegiatan SQA mengacu pada aspek fungsional, manajerial, dan ekonomi pada pengembangan dan pemeliharaan perangkat lunak. Kegiatan ini dilakukan untuk mencegah, mendeteksi, dan memperbaiki penyebab kesalahan. Tujuan dari kegiatan SQA dilihat dari aspek-aspek sebagai berikut [Galin]:

1. Pengembangan Perangkat Lunak (*Process*-*Oriented*):

* Menjamin tingkat keyakinan yang dapat diterima bahwa perangkat lunak akan sesuai dengan kebutuhan teknis fungsional.
* Menjamin tingkat keyakinan yang dapat diterima bahwa perangkat lunak akan sesuai untuk penjadwalan manajerial dan kebutuhan anggaran.
* Memulai dan mengelola kegiatan untuk perbaikan dan efisiensi pengembangan perangkat lunak yang lebih besar. Hal ini dapat meningkatkan prospek kebutuhan fungsional dan manajerial yang akan dicapai serta mengurangi biaya pengembangan perangkat lunak dan kegiatan SQA.

2. Pemeliharaan Perangkat Lunak (Product-Oriented):

* Menjamin dengan tingkat keyakinan yang dapat diterima bahwa kegiatan pemeliharaan perangkat lunak akan sesuai dengan kebutuhan teknis fungsional.
* Menjamin dengan tingkat keyakinan yang dapat diterima bahwa kegiatan pemeliharaan perangkat lunak akan sesuai dengan penjadwalan manajerial dan kebutuhan anggaran.
* Memulai dan mengelola kegiatan untuk memperbaiki serta meningkatkan efisiensi pemeliharaan perangkat lunak dan kegiatan SQA. Hal ini dapat meningkatkan prospek pencapaian kebutuhan fungsional dan manajerial serta mengurangi biaya.
  1. ***Quality Models***

Software Quality Assurance (SQA) memiliki beberapa pilihan model kulitas yang dapat digunakan untuk mengukur jaminan kualitas perangkat lunak. Biasanya, software yang berkualitas memiliki identifikasi sebagai berikut [D. Milicic]:

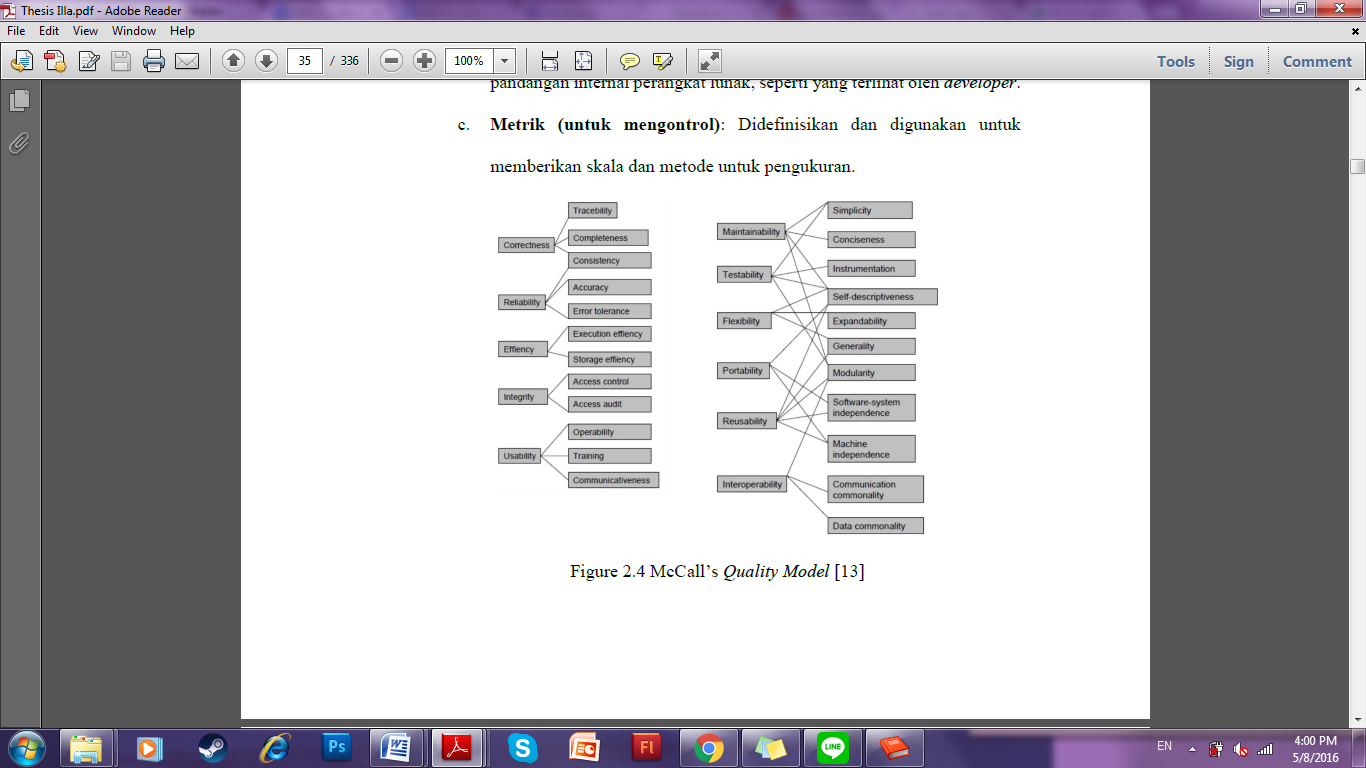
1. Kesesuaian dengan Spesifikasi: Kualitas yang didefinisikan sebagai produk dan jasa yang terukur serta karakteristik yang memenuhi spesifikasi yang telah ditetapkan sebelumnya.
2. Kebutuhan Pelanggan: Kualitas yang diidentifikasi terlepas dari karakteristik terukur. Hal ini mendefinisikan kualitas sebagai kemampuan produk atau jasa untuk memenuhi harapan pelanggan secara eksplisit atau tidak.

Model kualitas yang memiliki standar internasional dapat dijadikan acuan yang terpercaya saat proses pengukuran kualitas perangkat lunak dilakukan. Model kualitas yang berstandar internasional dan paling umum digunakan antara lain McCall’s Quality Model (1977), Boehm’s Quality Model (1978), dan ISO 9126’s Quality Model (2001) [Milicic 13].

**2.3.1 McCall’s *Quality Model***

Pendiri model kualitas yang masih terkenal dari model kualitas saat ini adalah model kualitas yang disampaikan oleh Jim McCall (juga dikenal sebagai General Electrics Model of 1977). Model kualitas McCall mencoba untuk menjembatani kesenjangan antara pengguna dan developer dengan berfokus pada sejumlah faktor kualitas software terhadap pandangan pengguna dan prioritas developer. Model ini berasal dari militer AS (dikembangkan untuk US Air Force, dipromosikan dalam DoD) dan ditujukan kepada pengembang sistem dan proses pengembangan sistem. [Milicic 13]

Model kualitas McCall memiliki tiga perspektif utama untuk mendefinisikan dan mengidentifikasi kualitas produk perangkat lunak, antara lain revisi produk, transisi produk, dan operasi produk. Revisi produk adalah kemampuan untuk mengalami perubahan, transisi produk adalah kemampuan beradaptasi dengan lingkungan baru, dan operasi produk adalah kemampuan dalam mengoperasikan produk. Revisi produk meliputi pemeliharaan (*maintainability*), *flexibility*, dan *testability*. Transisi produk mencakup hal-hal yang berkaitan dengan *portability*, *reusability*, dan *interoperability*. Sementara, kualitas operasi produk berkaitan pada kebenaran (*correctness*), kehandalan (*reliability*), efisiensi, integritas, dan kegunaan (*usability*). [Milicic 13]



Pada model kualitas McCall, seperti yang terlihat pada Figure 2.4, tiga jenis karakteristik kualitas (perspektif utama) dirinci dalam hierarki faktor, kriteria, dan metrik sebagai berikut [Milicic 13]:

1. 11 faktor (untuk menentukan): Menggambarkan pandangan eksternal dari perangkat lunak, seperti yang dilihat oleh pengguna.

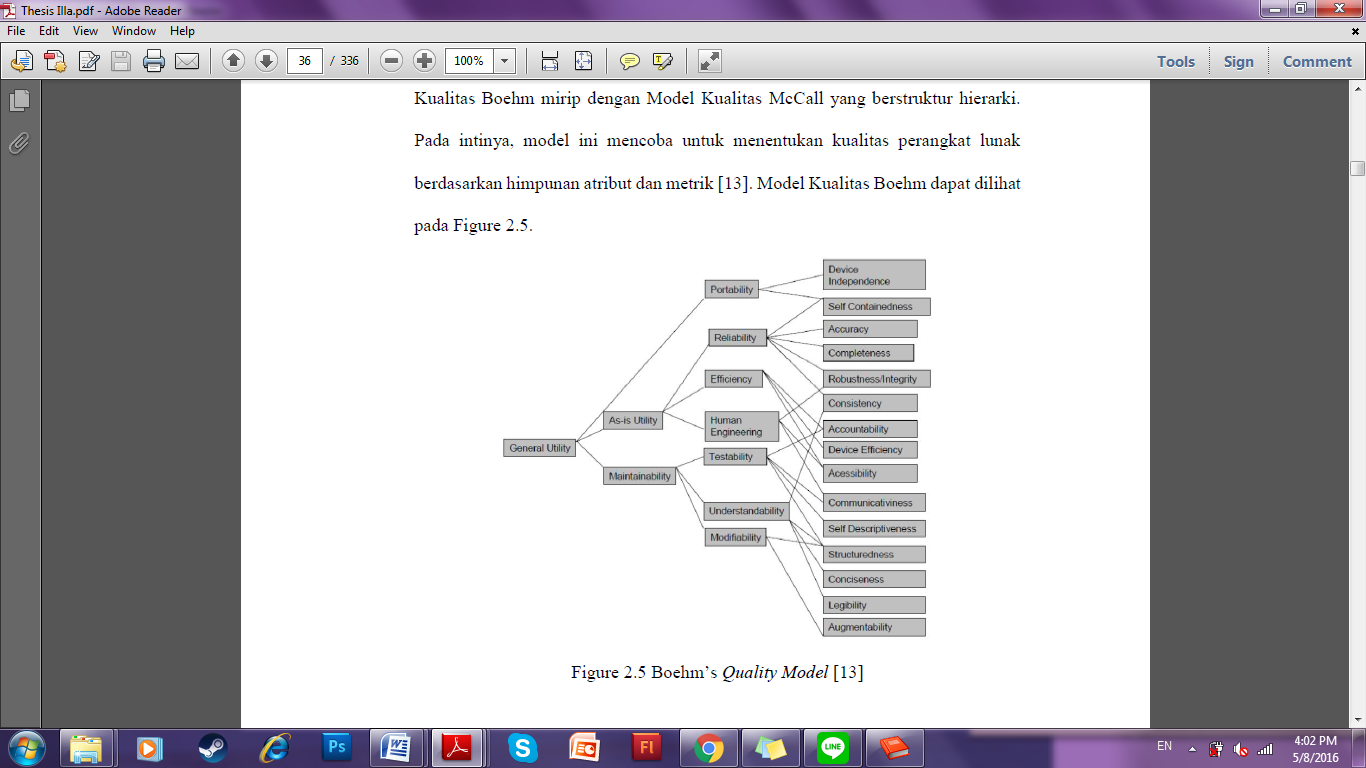
2. 23 kriteria kualitas (untuk membangun): Menggambarkan pandangan internal perangkat lunak, seperti yang terlihat oleh developer.

3. Metrik (untuk mengontrol): Didefinisikan dan digunakan untuk memberikan skala dan metode untuk pengukuran.

Model Kualitas McCall pada Figure 2.4 menjelaskan bahwa hierarki 11 faktor kualitas berada di sisi kiri dan 23 kriteria kualitas berada di sisi kanan. Ide di balik Model Kualitas McCall adalah faktor kualitas harus memberikan gambaran kualitas perangkat lunak yang lengkap. [Milicic 13]

**2.3.2 Boehm’s *Quality Model***

Model kualitas Boehm adalah model kualitas yang disampaikan pada tahun 1978 oleh Barry W. Boehm dengan membahas kekurangan kontemporer model yang secara otomatis dan kuantitatif mengevaluasi kualitas perangkat lunak. Model Kualitas Boehm mirip dengan Model Kualitas McCall yang berstruktur hierarki. Pada intinya, model ini mencoba untuk menentukan kualitas perangkat lunak berdasarkan himpunan atribut dan metrik [Milicic 13]. Model Kualitas Boehm dapat dilihat pada Figure 2.5.



Model berstruktur hierarki yang terlihat pada Figure 2.5 memiliki tingkatan karakteristik mulai dari karakteristik tingkat tinggi hingga rendah. Karakteristik tingkat tinggi merupakan persyaratan tingkat tinggi dasar pada penggunaan aktual yang mengevaluasi kualitas perangkat lunak. Karakteristik tingkat tinggi membahas tiga pertanyaan utama yang biasanya diajukan oleh seorang pembeli perangkat lunak, seperti [Milicic 13]:

1. *As-is Utility*: Seberapa baik (mudah, handal, efisien) perangkat lunak ini dapat saya gunakan?
2. *Maintainability*: Seberapa mudah untuk memahami, memodifikasi, dan menguji ulang?
3. *Portability*: Dapatkah saya masih menggunakannya jika saya mengubah lingkungan saya?

Karakteristik tingkat menengah memiliki tujuh faktor kualitas Boehm yang mewakili kualitas yang diharapkan dari sebuah sistem perangkat lunak. Ketujuh faktor kualitas Boehm adalah sebagai berikut [Milicic 13]:

1. *Portability* (*General utility characteristics*): Kode yang memiliki karakteristik portabilitas sampai batas yang dapat dioperasikan dengan mudah dan baik pada konfigurasi komputer lain dari konfigurasi saat ini.
2. Kehandalan (*As-is utility characteristics*): Kode yang memiliki karakteristik kehandalan sampai batas yang dapat diharapkan dalam melakukan fungsinya.
3. Efisiensi (*As-is utility characteristics*): Kode yang memiliki karakteristik efisiensi sampai batas yang dapat memenuhi tujuannya tanpa pemborosan sumber daya.
4. Kegunaan (*As-is utility characteristics, Human engineering*): Kode yang memiliki karakteristik kegunaan sampai batas yang dapat dihandalkan, efisien, dan human-engineered.
5. *Testability* (*Maintainability* *characteristics*): Kode yang memiliki karakteristik testability sampai batas yang dapat memfasilitasi pembentukan kriteria verifikasi dan mendukung evaluasi kinerjanya.
6. *Understandability* (*Maintainability* *characteristics*): Kode yang memiliki karakteristik yang saling memahami sampai batas yang tujuannya jelas untuk inspektur.
7. *Flexibility* (*Maintainability* *characteristics*, *Modifiability*): Kode yang memiliki karakteristik modifiability sampai batas yang memfasilitasi penggabungan perubahan setelah sifat perubahan yang diinginkan telah ditentukan.

Struktur tingkat terendah dari karakteristik hierarki dalam Model Boehm adalah hierarki metrik karakteristik primitif. Karakteristik primitif memberikan dasar untuk menentukan kualitas metrik yang merupakan salah satu pencapaian saat membangun Model Kualitas Boehm. Meskipun Boehm dan McCall memiliki kemiripan, Model Boehm lebih berfokus pada upaya efektivitas biaya pemeliharaan perangkat lunak. [Milicic 13]

**2.3.3 ISO 9126’s Quality Model**

*International Organization for Standardization* (ISO) dan *International Electrotechnical Commission* (IEC) telah menetapkan standar ISO/IEC yang berkaitan dengan kualitas perangkat lunak. ISO yang pertama kali adalah ISO 9000 yang memiliki 3 pedoman untuk menerapkan standar ISO 9001 yang berkaitan dengan proses jaminan kualitas. Proses ini dilakukan untuk proses pengembangan, penyediaan, instalasi, dan pemeliharaan perangkat lunak komputer. Kemudian, standar ISO/IEC 9126 ditetapkan untuk kualitas produk perangkat lunak dan standar ISO/IEC 14598 untuk evaluasi produk perangkat lunak. Standar lainnya yang dapat digunakan bersama dengan ISO/IEC 9126 dan ISO/IEC 14598 adalah sebagai berikut [14 P. Botella]:

1. ISO/IEC 12119 - Persyaratan kualitas untuk paket perangkat lunak.

2. ISO/IEC 12207 - Proses siklus hidup *software*.

3. ISO/IEC 14143 - Pengukuran *software*.

4. ISO/IEC 15271 - Panduan untuk ISO/IEC 12207

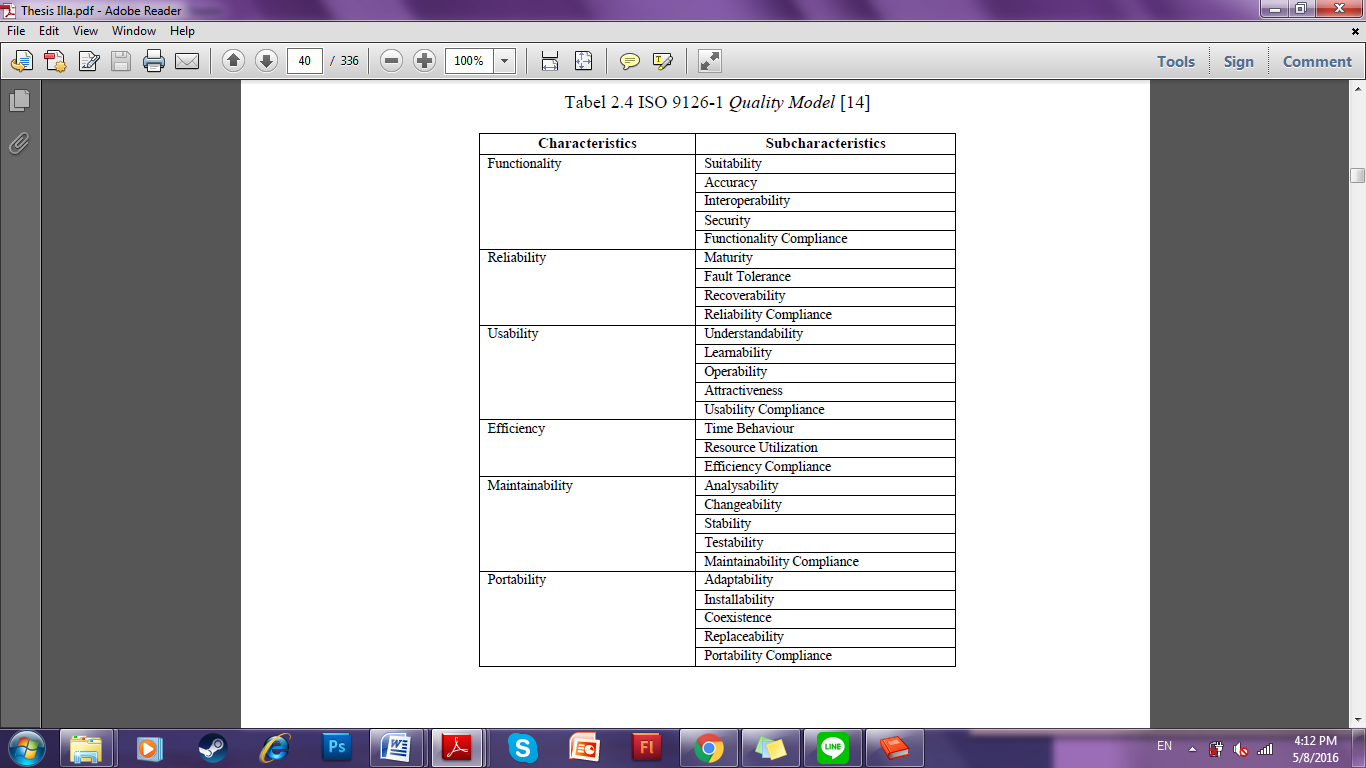
5. ISO/IEC 15504 - Penilaian proses *software* (juga dikenal sebagai SPICE

– Software Process Improvement for Capability Determination).

6. ISO/IEC 15939 - Proses pengukuran software.

Standar ISO/IEC 9126 membuat suatu perbedaan antara kualitas internal dan kualitas eksternal. Model ini mengkategorikan atribut kualitas perangkat lunak menjadi karakteristik. Atribut yang dapat diukur selama proses pengembangan disebut sebagai internal. Sementara, perilaku eksternal dapat diukur selama proses pengujian dan kualitas pandangan pengguna [14 P. Botella]. ISO/IEC 9126 terdiri dari empat bagian yang meliputi ISO 9126-1 *Quality Model*, ISO 9126-2 *External Metrics*, ISO 9126-3 *Internal Metrics*, serta ISO 9126-4 *Quality in Use Metrics* [13 Milicic].

Model kualitas ISO 9126-1 didefinisikan melalui karakteristik umum perangkat lunak, yang selanjutnya disempurnakan ke dalam subcharacteristics, yang pada gilirannya diuraikan ke dalam atribut dan menghasilkan sebuah hierarki bertingkat. Ide utama di balik standar ini adalah definisi model kualitas dan penggunaannya sebagai kerangka kerja untuk evaluasi perangkat lunak. Pada ISO 9126-1 *Quality Model* versi 2001 terdapat enam karakteristik dan 27 subkarakteristik yang dijabarkan pada Tabel 2.4. [P. Botella 14]



Pada versi 2001 ini terdapat penambahan subkarakteristik *Compliance* di setiap karakteristik ISO 9126-1 Quality Model. Penjelasan rinci keenam karakteristik ISO 9126-1 Quality Model adalah sebagai berikut:

1. *Functionality*: Kemampuan perangkat lunak dalam menyediakan fungsi yang sesuai dengan kebutuhan pengguna ketika digunakan dalam kondisi tertentu.

2. *Reliability*: Kemampuan perangkat lunak dalam mempertahankan tingkat kinerja ketika digunakan dalam kondisi tertentu.

3. *Usability*: Kemampuan perangkat lunak yang berhubungan dengan penggunaan perangkat lunak yang dilakukan oleh pengguna.

4. *Efficiency*: Kemampuan perangkat lunak dalam memberikan tingkat kinerja yang sesuai dan jumlah sumber daya yang digunakan ketika perangkat lunak dijalankan.

5. *Maintainability*: Kemampuan perangkat lunak yang berhubungan dengan upaya yang diperlukan untuk dimodifikasi atau diubah.

6. *Portability*: Kemampuan perangkat lunak untuk dikirim ke lingkungan yang berbeda atau satu lingkungan yang lain.

**2.3.4 ISO 25010 *Quality Model***

ISO/IEC 25010:2010 (ISO 25010), adalah bagian dari rangkaian yang dikenal dengan *Software Quality Requirements and Evaluation* (SQuaRE), mendefinisikan kualitas sistem sebagai “*the degree to which the system satisfies the stated and implied needs of its various stakeholders, and thus provides value*” [ISO].

ISO 25010 merupakan pengembangan dari ISO 9126. ISO 25010 memiliki dua dimensi utama: Quality-in-use (QinU) dan Product Quality. QinU mendefinisikan karakteristik berhubungan pada interaksi manusia dengan sistem sedangkan Product quality mendefinisikan karakteristik dasar sebuah produk. QinU didefinisikan sebagai “*capability of a software product to influence users’ effectiveness, productivity, safety and satisfaction to satisfy their actual needs when using the software product to achieve their goals in a specified context of use*” [ISO 2].

Model QinU terdiri dari 5 karakteristik: *effecticeness, efficiency, satisfaction, freedom from risk dan context coverage*. Figure 5 menjabarkan karakteristik dan sub-karakteristik yang terdapat pada dimensi QinU [ISO].

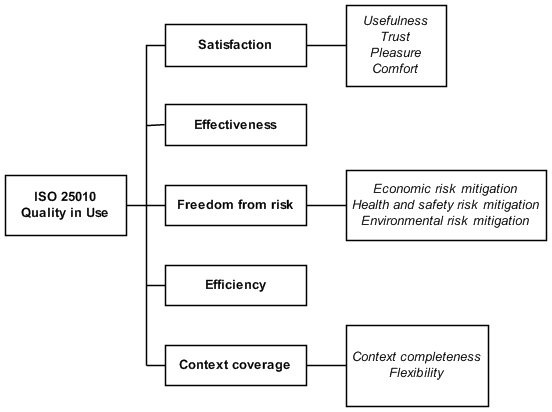


Figure 5: ISO 25010 Quality in Use

Definisi dari masing-masing karakteristik dapat dilihat pada tabel 1 berikut.

Tabel 1: Quality in Use characteristics

|  |  |
| --- | --- |
| ***Characteristic*** | ***Definition*** |
| ***Effectiveness*** | *Accuracy and completeness with which users achieve specified goals (ISO 9241-11).* |
| ***Efficiency*** | *Resources expended in relation to the accuracy and completeness with which users achieve goals (ISO 9241-11).* |
| ***Freedom From Risk*** | *Degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment.* |
| ***Satisfaction*** | *Degree to which user needs are satisfied when a product or system is used in a specified context of use.* |
| ***Context Coverage*** | *Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified.* |

Dimensi Product Quality terdiri dari delapan karakteristik kualitas, yaitu *Funtional suitability, Performance efficiency, Compatibility, Usability, Reliability, Security, Maintainability dan Portabilit*y. Figure 6 menjabarkan karakteristik dan sub-karakteristik pada dimensi *Product quality*.

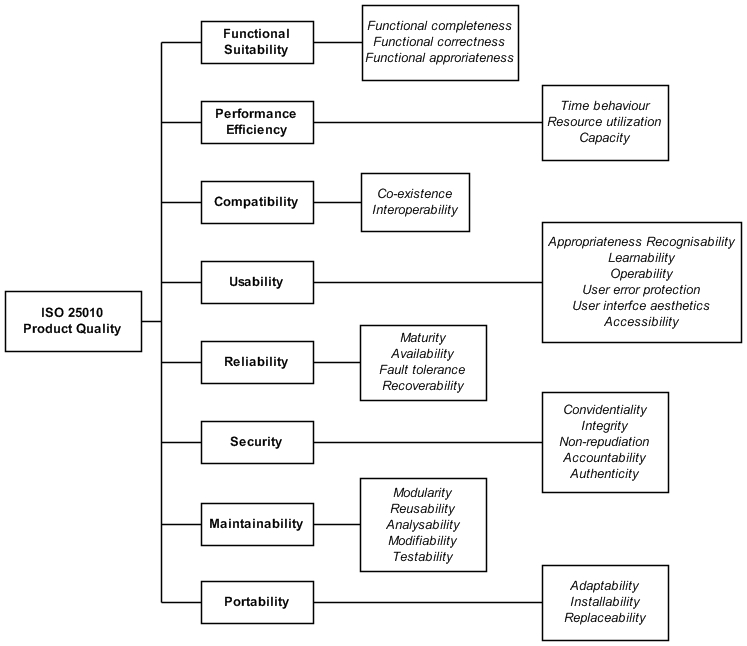


Figure 6: ISO 25010 Product quality

Definisi dari masing-masing karakteristik dapat dilihat pada tabel 2 sebagai berikut.

Tabel 2: *Product qality characteristics*

|  |  |
| --- | --- |
| ***Characteristic*** | ***Definition*** |
| ***Functional completeness*** | *Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.* |
| ***Performance efficiency*** | *Represents the performance relative to the amount of resources used under stated conditions.* |
| ***Compatibility*** | *Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.* |
| ***Usability*** | *Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.* |
| ***Reliability*** | *Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.* |
| ***Security*** | *Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.* |
| ***Maintainability*** | *Degree of effectiveness and efficiency with which a product or system can be modified to omporve it, correct it or adapt it to changes in environment, and in requirement.* |
| ***Portability*** | *Degree of effectiveness and efficiency with which a system, pproduct or component can be transferred from one hardware, software or other operational or usage environment to another.* |

**2.3.5 Perbandingan Model Kualitas**

Perbandingan model kualitas McCall, Boehm, ISO 9126 dan ISO 25010 berdasarkan karakteristiknya dijabarkan dalam tabel 3.

Tabel 3: Perbandingan Model Kualitas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Faktor Kualitas** | **Mc Call (1977)** | **Boehm (1978)** | **ISO 9126 (2000)** | **ISO 25010 (2010)** |
| 1 | Accuracy |  |  | X | X |
| 2 | Adaptability |  |  |  | X |
| 3 | Analyzability |  |  | X | X |
| 4 | Attractiveness |  |  | X | X |
| 5 | Changeability |  |  | X | X |
| 6 | Correctness | X |  |  | X |
| 7 | Efficiency | X | X | X | X |
| 8 | Flexibility | X |  |  |  |
| 9 | Functionality |  |  | X | X |
| 10 | Human Engineering |  | X |  |  |
| 11 | Installability |  |  | X | X |
| 12 | Integrity | X |  |  | X |
| 13 | Interoperability | X |  |  | X |
| 14 | Maintainability | X |  | X | X |
| 15 | Maturity |  |  | X | X |
| 16 | Modifiability |  |  |  | X |
| 17 | Operability |  |  | X | X |
| 18 | Performance |  |  | X | X |
| 19 | Portability | X | X | X | X |
| 20 | Reliability | X | X | X | X |
| 21 | Resource Utilization |  |  | X | X |
| 22 | Reusability | X |  |  | X |
| 23 | Stability |  |  | X | X |
| 24 | Suitability |  |  | X | X |
| 25 | Supportability |  |  | X | X |
| 26 | Testability | X | X | X | X |
| 27 | Transferability |  |  |  | X |
| 28 | Understandability |  | X | X | X |
| 29 | Usability | X |  | X | X |

Berdasarkan tabel 3 Model ISO 25010 memiliki karakteristik yang paling lengkap dibanding model kualitas lainnya karena mencakup 26 dari 28 fitur yang ada. Dari perbandingan tersebut, karakteristik *efficiency*, *portability* dan *reliability* merupakan karakterstik yang muncul di semua *quality model*. *Quality model* ISO 25010 merupakan pengembangan dari model kualitas ISO 9126,  sehingga dapat dilihat pada tabel 3 model ISO 9126 memiliki karakteristik yang hampir lengkap seperti pada ISO 25010. Menurut penelitian [P. Botella 14] ISO 9126 memiliki beberapa keterbatasan karena sifat generiknya. Beberapa konsep yang disajikan oleh ISO 9126 perlu disempurnakan sebelum diterapkan benar-benar dalam sebuah proyek. Selain itu, unsur-unsur dari metriks perangkat lunak tidak jelas ketika mendefinisikan standar [Franca].

Karakteristik baru telah dimasukkan dalam ISO 25010 adalah *security* dan *compatibility*. Kedua karakteristik tersebut tidak disajikan dalam ISO 9126. Selain itu hirarki karakteristik dan sub-karakteristik di susun ulang dengan tujuan meningkatkan pemahaman mengenai konsep terkait. Hal tersebut dilakukan untuk mengatasi keterbatasan ISO 9126 sehubungan dengan sifat abstrak, ketidaklengkapan dan ketidakjelasan seperti yang dipaparkan oleh peneliti [Al-Kilidar]. Salah satu alasan lain model ISO 9126 sudah tidak relevan untuk dijadikan standar dalam pengujian perangkat lunak karena perubahan ICT yang sangat cepat dan sangat berbeda dari satu decade yang lalu. Evolusi dalam dunia ICT seperti penggunaan *memory* yang lebih besar, tampilan yang lebih baik hingga prosesor yang lebih cepat memungkinkan pengembangan sistem aplikasi baru yang juga membutuhkan kualitas yang berbeda seperti dengan menggunakan model kualitas ISO 25010 [5 Veenendaal]. Berdasarkan data dari penelitian dan perbandingan model kualitas tersebut, penelitian ini menggunakan model ISO 25010 untuk mengukur kualitas dari *transportation service application* yang memiliki jumlah unduhan tertinggi, yaitu GO-JEK dan GRAB.

* 1. **Metode Penentuan Bobot**

Bobot biasanya ditentukan untuk memperoleh urutan kriteria mulai dari level yang paling diprioritaskan hingga level terendah. Penentuan bobot dilakukan dengan tujuan mempermudah pengambilan keputusan menggunakan metode-metode yang ada. Dari banyaknya metode penentuan bobot, dua di antaranya adalah metode *Analytical Hierarchy Process* (AHP) dan *Simple Additive* *Weighting* (SAW).

**2.4.1 *Analytical Hierarchy Process* (AHP)**

*Analytical Hierarchy Process* (AHP) merupakan model pengambilan keputusan yang dikembangkan oleh Thomas L. Saaty pada tahun 1980. Model ini telah berhasil diterapkan pada berbagai situasi pengambilan keputusan. AHP tidak hanya berperan sebagai pengambil keputusan tunggal untuk memilih alternatif yang paling cocok, tetapi diperluas untuk pengambilan keputusan kelompok. AHP akan mempersatukan penilaian preferensi pengambil keputusan untuk setiap keputusan alternatif di bawah setiap kriteria dalam hierarki keputusan. Langkah-langkah untuk memperoleh keputusan pada metode AHP adalah sebagai berikut [AHP]:

1. Menguraikan masalah dalam struktur hierarki keputusan, termasuk semua kriteria, sub-kriteria, dan alternatif keputusan.
2. Melakukan perbandingan berpasangan dari semua alternatif keputusan di bawah setiap kriteria berdasarkan skala penilaian preferensi Saaty yang dijabarkan pada Table 2.6.

Tabel 2.6. Intensitas Perbandingan Berpasangan AHP [AHP]

|  |  |  |
| --- | --- | --- |
| **Intensitas** | **Definisi** | **Penjelasan** |
| 1 | Sama Penting | Kedua elemen memiliki pengaruh yang sama penting. |
| 3 | Sedikit Lebih Penting | Elemen yang satu sedikit lebih penting daripada elemen lainnya. |
| 5 | Lebih Penting | Elemen yang satu lebih penting daripada elemen lainnya. |
| 7 | Sangat Penting | Elemen yang satu sangat lebih penting daripada elemen lainnya. |
| 9 | Mutlak Lebih Penting | Elemen yang satu mutlak lebih penting daripada elemen lainnya. |
| 2,4,6,8 | Mewakili nilai-nilai menengah diantara kedua elemen. | |
| Kebalikan | Jika aktivitas i mendapat satu angka dibanding dengan aktivitas j, maka j memiliki nilai kebalikannya dengan i. | |

1. Mengambil bobot prioritas lokal dengan menggunakan metode eigenvector atau metode pendekatan lainnya.
2. Satukan bobot prioritas lokal untuk menghasilkan keseluruhan bobot preferensi pada setiap alternatif keputusan.
3. Periksa tingkat perbandingan berpasangan yang tidak konsisten. Apabila tingkat yang tidak konsisten ditolak, maka pengambil keputusan harus merevisi perbandingan berpasangan (lihat Langkah 2).

Untuk melakukan perbandingan berpasangan dari semua kriteria yang relevan, Saaty mengusulkan pembentukan perbandingan berpasangan *n x n* pada matriks A = {*aij*}. Matriks A memiliki nilai-nilai *aij* sesuai dengan skala penilaian yang disajikan pada Table 2.6. [AHP]

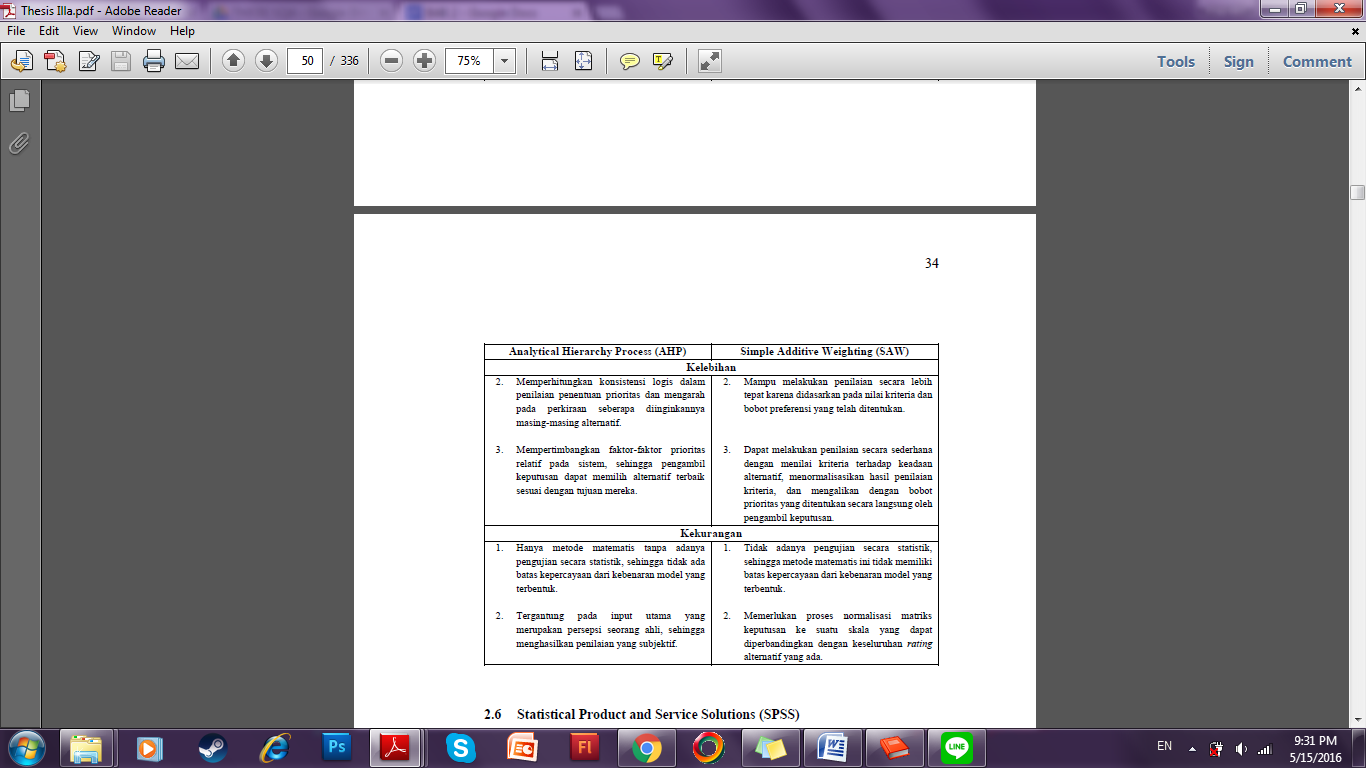
**2.4.2 *Simple Additive* *Weighting* (SAW)**

Metode ini merupakan metode yang paling dikenal dan paling banyak digunakan orang dalam menghadapi sitasi MCDM (Multiple Criteria Decision Making). Metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Metode SAW lebih berfokus dalam melakukan penilaian secara lebih tepat. Hal ini didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan. Selain itu, SAW juga dapat menyeleksi alternatif terbaik dari sejumlah alternatif yang ada karena adanya proses perankingan setelah menentukan nilai bobot untuk setiap atribut [SAW]. Langkah-langkah penyelesaian menggunakan metode SAW adalah sebagai berikut:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan yaitu, *Ci*.
2. Menentukan rating kecocokan setiap alternative pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (*Ci*), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut sehingga diperoleh matrks ternomalisasi *R*.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumalahan dari perkalian matriks ternomalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternative terbaik (*Ai*) sebagai solusi.

**2.4.3 Perbandingan Metode AHP dengan SAW**

Metode *Analytical Hierarcy Proces* dan *Simple Additiva Weighting* merupakan metode ang digunkn untuk menyelesaikan masalah pengamblan keputusan. Kedua metode ini memiliki konsep yang berbeda, meskipun tujuannya selaras. Metode AHP dan SAW memiliki kelebihan dan kekurangaan masing-asing yang dijabarkan pada tabel 6 berikut.



* 1. **Statistical Product and Service Solutions (SPSS)**

SPSS Statistik adalah sistem yang komprehensif untuk menganalisis data. SPSS Statistik dapat mengambil data hampir dari semua jenis file. Data tersebut digunakan untuk menghasilkan laporan, grafik, bidang distribusi dan tren, statistik deskriptif, dan analisis statistik yang kompleks. Kepemilikan SPSS Statistik dipegang oleh International Business Machines (IBM) Corporation. Ada beberapa jenis window yang umum digunakan pada semua versi SPSS Statistik, di antaranya [SPSS]:

1. **Data Editor**, menampilkan isi dari file data. Pengguna dapat membuat file data baru atau memodifikasi file data yang ada pada Data Editor.

2. **Viewer,** semua statistik hasil, tabel, dan grafik ditampilkan dalam Viewer. Pengguna dapat mengedit output dan menyimpannya untuk digunakan nanti. Viewer window akan terbuka secara otomatis saat pertama kali pengguna menjalankan prosedur yang menghasilkan output.

3. **Pivot Table Editor**, output yang ditampilkan dalam tabel pivot dapat dimodifikasi dengan berbagai cara menggunakan Pivot Table Editor. Pengguna dapat mengedit teks, data swap baris dan kolom, menambah warna, membuat tabel multidimensi, serta selektif dalam menyembunyikan dan menunjukkan hasil.

4. **Chart Editor**, pengguna dapat memodifikasi grafik resolusi tinggi dan plot pada grafik window. Pengguna dapat mengubah warna, pilih jenis font atau ukuran yang berbeda, beralih dari sumbu horizontal dan vertikal, memutar 3-D scatterplots, serta mengubah jenis grafik.

5. **Text Output Editor**, text output yang tidak ditampilkan dalam tabel pivot dapat dimodifikasi dengan Text Output Editor. Pengguna dapat mengedit karakteristik output dan perubahan font (jenis, gaya, warna, ukuran).

6. **Syntax Editor**, pengguna dapat menyisipkan pilihan kotak dialog ke Syntax window, di mana pilihan pengguna muncul dalam bentuk sintaks perintah. Pengguna kemudian dapat mengedit sintaks perintah untuk menggunakan fitur khusus yang tidak tersedia melalui kotak dialog. Selain itu, pengguna juga dapat menyimpan perintah dalam file untuk digunakan pada sesi berikutnya.

Pada penelitian ini, SPSS yang akan digunakan adalah SPSS versi 20 (masih belom pasti) untuk menguji validitas dan reliabilitas dari hasil kuesioner yang telah diberikan oleh responden. Versi 20 ini menyediakan render yang lebih cepat dari tabel pivot. Tabel pivot pada versi 20 jauh lebih cepat daripada versi sebelumnya, tetapi tetap mempertahankan dukungan penuh untuk pivoting dan editing. Jika pengguna menggunakan render lightweight table yang cepat dalam versi 19, pengguna akan menemukan hasil yang sebanding untuk tabel pivot dalam versi 20. Pengguna yang membutuhkan kompatibilitas dengan versi sebelumnya ke versi 20 dapat memilih untuk menghasilkan tabel warisan (disebut sebagai full-featured table dalam versi 19). [SPSS]

* 1. **Penelitian Terdahulu**