

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Untuk mendukung penelitian ini, perlu dikemukakan tinjauan pustaka dari penelitian sebelumnya yang berkaitan dengan ruang lingkup pembahasan, digunakan sebagai penunjang dan landasan penelitian yang sedang dilakukan. Berikut tinjauan pustaka yang telah dilakukan sebelumnya oleh peneliti terdahulu :

2.1.1 Tinjauan Terhadap Literatur 01

Penelitian oleh Sindhu and Sagar, (2017) dari *IOP Conference Series: Materials Science and Engineering, Research Scholar R.V College of Engineering, India*. dengan judul “Dictionary Based Machine Translation from Kannada to Telugu”. Dalam penelitian tersebut penulis mengembangkan teknologi mesin penerjemah berfokus pada terjemahan mesin berbasis Kamus untuk Bahasa *Kannada* ke *Telugu*. Metode yang diusulkan menggunakan kamus untuk menerjemahkan kata demi kata tanpa banyak korelasi semantik. Proses mesin penerjemahan berbasis kamus ini memiliki sub proses sebagai berikut: *Morph analyzer*, kamus, transliterasi, pertukaran tata bahasa, dan generator morph. Sebagai bagian dari penelitian kamus dwibahasa dengan 8000 entri kata yang dikembangkan dengan tabel pemetaan sufiks. Sistem ini diuji untuk anak-anak. Dalam waktu dekat sistem ini dapat lebih ditingkatkan dengan mendefinisikan peraturan pertukaran tata bahasa. (Sindhu and Sagar, 2017)

2.1.2 Tinjauan Terhadap Literatur 02

Penelitian oleh Afifah, Santoso dan Yuliana, (2010) dari jurusan Teknik Telekomunikasi, Politeknik Elektronika Negeri Surabaya dengan judul “Pembuatan Kamus Elektronik Kalimat Bahasa Indonesia dan Bahasa Jawa untuk Aplikasi Mobile Menggunakan *Interpolation Search*”. Penelitian yang dilakukan oleh penulis adalah salah satu potensi dari aplikasi mesin penerjemah, yaitu pembuatan *software* kamus elektronik atau translator J2J untuk menerjemahkan kalimat dari bahasa Indonesia ke bahasa Jawa atau sebaliknya. Pembuatan *software* ini menggunakan pemrograman J2ME. Proses penerjemahan kalimat dimulai dari pemecahan kalimat dengan pembacaan tanda spasi, pencarian padanan kata dengan metode *interpolation search* dan penggabungan kata menjadi kalimat kembali. Hasil penelitian yang diperoleh dari pengujian penerjemahan kalimat bahasa Indonesia ke bahasa Jawa untuk jumlah kata sama dengan 5, diperlukan rata-rata waktu 12.24 *milisecond* pada emulator, dan 52.8 *milisecond* pada ponsel, sedangkan dari bahasa Jawa ke Indonesia diperlukan rata-rata waktu 11.68 *milisecond* pada emulator, dan 38.48 *milisecond* pada ponsel. Selain itu dari kuisisioner didapat data bahwa 100% responden mengatakan bahwa aplikasi J2J ini membantu dalam menerjemahkan bahasa Jawa dimana aplikasi ini bisa dijalankan dimanapun, dan kapanpun dengan media perangkat *mobile* yang praktis. (Afifah, Santoso and Yuliana, 2010)

2.1.3 Tinjauan Terhadap Literatur 03

Penelitian oleh Holis, (2019) dari Fakultas Teknologi Informasi, Universitas KH. A. Wahab Hasbullah dengan judul “Aplikasi Kamus Bahasa Madura Berbasis

Website”. Penelitian yang dilakukan oleh penulis adalah salah satu potensi dari aplikasi mesin penerjemah, yaitu pembuatan *software* untuk menerjemahkan kalimat dari bahasa madura ke bahasa Indonesia atau sebaliknya. Aplikasi kamus terjemahan ini menggunakan algoritma Pencarian Sekuensial untuk pemrosesan pencarian kata. Metode pencarian ini sangat cocok digunakan untuk aplikasi kamus. Sistem yang dikembangkan berfokus pada bagaimana menyediakan fasilitas untuk menerjemahkan kata/kalimat dari Bahasa Indonesia ke Bahasa Madura maupun dari Bahasa Madura ke Bahasa Indonesia dengan kesimpulan aplikasi kamus telah berhasil dalam melakukan terjemahan baik dalam bentuk kata ataupun kalimat dan mempermudah untuk memahami cara pengucapan kata karena aplikasi ini memiliki terjemahan suara.

2.1.4 Tinjauan Terhadap Literatur 04

Penelitian oleh Imam Santoso, (2017) dari jurusan teknik informatika, Fakultas Teknik Universitas Negeri Yogyakarta dengan judul “Pengembangan Aplikasi Penerjemah Bahasa Indonesia Bahasa Jawa Pada Platform Berbasis Android”. Penelitian ini dirancang untuk mengembangkan aplikasi penerjemah Bahasa Indonesia dan Bahasa Jawa dengan *unggah-ungguh basa* pada platform Android, dan mengetahui kualitas aplikasi penerjemah Bahasa Indonesia-Bahasa Jawa berdasarkan standar pengujian kualitas perangkat lunak ISO 25010 meliputi aspek *functional suitability*, *performance efficiency*, *compatibility*, *usability*, *reliability*, dan materi. Hasil dari penelitian ini adalah pengembangan aplikasi penerjemah Bahasa Indonesia dan Bahasa Jawa dengan *unggah-ungguh basa* pada platform Android menggunakan Android Studio, hasil pengujian kualitas aplikasi diperoleh nilai *functional suitability* 100% (Sangat Layak), *performance efficiency*

menunjukkan skala kualitas "Layak", *compatibility* diperoleh skala kualitas "Sangat Layak", *usability* diperoleh nilai 80,04% (Layak), *reliability* diperoleh nilai 100% dengan skala kualitas "Sangat Layak", dan pengujian materi diperoleh nilai 100% (Sangat Layak).

2.1.5 Tinjauan Terhadap Literatur 05

Penelitian oleh Jafar Fakhrurozi dkk (2019) dari Fakultas Teknik Ilmu Komputer Universitas Teknokrat Indonesia dengan judul "Purwarupa Aplikasi Mesin Penerjemah Bahasa Lampung Dialek *Api* berbasis Kamus Bahasa Lampung Pemrograman *Python*". Penulis menggunakan pendekatan dengan cara kalimat bahasa Lampung akan dipecahkan berdasarkan spasi, hasilnya berupa daftar kata yang akan dilakukan pencocokan kata dengan kunci di *database* kamus. Hasil pencocokan tersebut adalah nilai, berupa kata dalam bahasa Indonesia, dari kunci *database* kamus untuk kemudian disusun kembali menjadi kalimat dalam bahasa Indonesia. Tujuan utama dari penelitian ini membangun purwarupa aplikasi mesin penerjemah bahasa Lampung dialek *api*. Purwarupa aplikasi yang dibuat sebagai rencana solusi alternatif dalam menerjemahkan bahasa Lampung dengan tetap mempertahankan penggunaan kamus. Dalam membangun aplikasi ini digunakan pendekatan metode *Waterfall* dengan fokus utama pada algoritma penerjemahan kalimat. Aplikasi dapat menerjemahkan kalimat bahasa Lampung sesuai dengan kosakata yang terdapat pada kamus rujukan yang digunakan.

2.2 Landasan Teori

2.2.1 Bahasa Lampung

Bahasa Lampung adalah bahasa daerah masyarakat di Provinsi Lampung. Pemerintah daerah terus menerus berupaya melakukan beragam cara guna pelestarian dan pemertahanan bahasa Lampung. Pemerintah daerah melalui Peraturan Gubernur nomor 39 tahun 2014 tentang Mata Pelajaran Bahasa dan Aksara Lampung menetapkan bahwa bahasa Lampung sebagai muatan lokal wajib pada jenjang satuan pendidikan dasar sampai menengah atas dan didukung oleh ketersediaan buku ajar mulai dari SD, SMP dan SMA, berikut kamus bahasa Lampung. Di Provinsi Lampung secara umum terbagi menjadi dua dialek utama yaitu dialek Api dan dialek Nyo. Pada penelitian ini hanya dilakukan pengamatan secara saksama terhadap bahasa Lampung dialek Api. Bahasa Lampung digunakan oleh masyarakat Lampung untuk berkomunikasi sehari-hari baik dalam lingkungan keluarga maupun pada acara adat. Bahasa Lampung termasuk dalam kelas *Austronesia* dalam keluarga bahasa Melayu Polinesia. Bahasa Lampung memiliki dua dialek utama yang hidup berdampingan dan keduanya aktif digunakan oleh masing-masing pengguna dialek tersebut. Dialek yang dimaksud adalah dialek A (api) dan dialek O (nyo) yang mengacu pada kata ‘apa’ (Megaria, 2013).

2.2.2 Karakteristik Bahasa Lampung

Bahasa Lampung mempunyai struktur tata bahasa yang serupa dengan Bahasa Indonesia. Di dalamnya terdapat subjek, predikat, objek, keterangan dan

lain-lain. Kalimat dalam bahasa Lampung juga serupa dengan kalimat dalam Bahasa Indonesia, ada kalimat tunggal, kalimat majemuk, kalimat tanya, kalimat perintah, kalimat tanya, kalimat berita dan lain-lain (Megaria, 2013).

Pada bagian ini disajikan berbagai kalimat dalam bahasa Lampung berikut terjemahannya dalam bahasa Indonesia (Megaria, 2013).

a. Kalimat tunggal

Contoh dalam bahasa Lampung '*Burhan lapah mit sekula*'.

b. Kalimat majemuk

Contoh dalam bahasa Lampung '*Burhan lapah mit sekula walau badan ni mak sihat*'.

c. Kalimat perintah

Contoh dalam bahasa Lampung '*Mejong pai!*'.

d. Kalimat tanya

Contoh dalam bahasa Lampung '*Ulah api sanak lunik miwang teghus?*'.

e. Kalimat berita

Contoh dalam bahasa Lampung '*Indui becawa, nyak mak haga lijung mit Jakarta*'.

f. Kalimat sempurna

Contoh dalam bahasa Lampung '*Nyak ngebattu ulun tuhani di ghani minggu*'.

2.2.3 Kamus

Secara Etimologi, kata *kamus* berasal dari kata dalam bahasa Arab, yaitu *qamus* (bentuk jamaknya *qawamus*). Bahasa Arab menyerap kata *kamus* dari kata

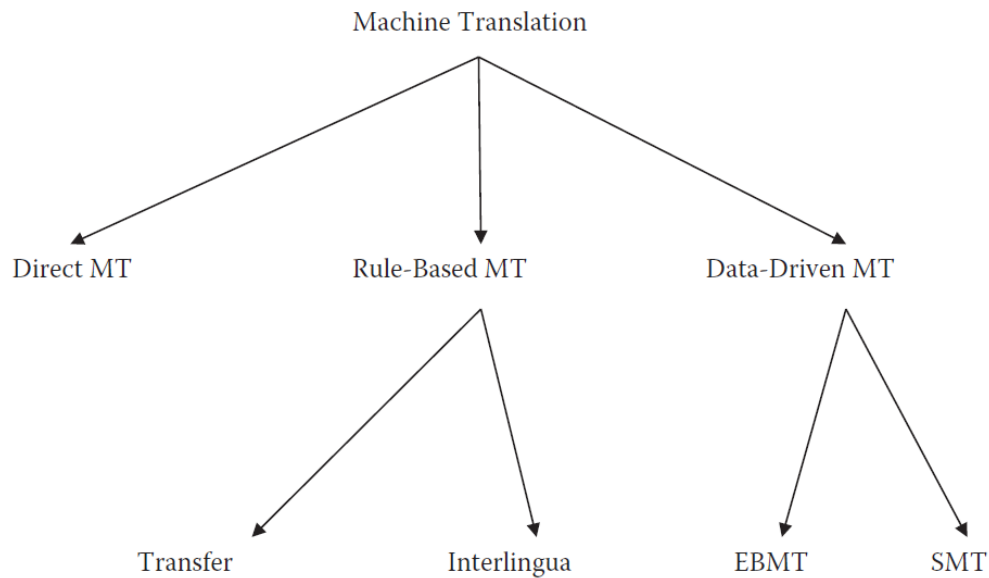
dalam bahasa Yunani kuno, *okeanos* yang berarti lautan. Tentu menjadi pertanyaan, bagaimana kata *kamus* yang berurusan dengan kosakata berasal dari bahasa Yunani kuno *okeanos* yang berarti *lautan*? Kalau kita mencoba untuk memahami sejarah kata itu, jelaslah bahwa kata *kamus* memiliki makna dasar *wadah pengetahuan*, khususnya pengetahuan bahasa yang tidak terhingga dalam dan luasnya, seluas dan sedalam lautan (Chaer, 2007:179)

2.2.4 Kamus Bahasa Lampung

Kamus yang digunakan dalam penelitian ini adalah kamus yang disusun oleh Herman, S.Pd.I (2013). Kamus Edisi ke 2 tahun 2013 ini merupakan salah satu kamus bahasa Lampung yang mewakili dialeg A dan dialeg O dalam satu buku, dan penggunaan kosa kata dalam kalimat menggunakan imbuhan dalam bahasa Lampung.

2.2.5 Mesin Penerjemah

Tiga paradigma utama pada mesin penerjemah yaitu *Direct machine translation* (DMT), *rule-based machine translation* (RBMT), *statistical machine translation* (SMT) dan *example-based machine translation* (EBMT). Pada bentuk aslinya dari ketiga paradigma tersebut, RBMT berbasis *rule* atau *knowledge*, SMT dan EBMT berbasis data yaitu korpus paralel. Ketiga paradigma tersebut ada pada gambar 2.1.



Gambar 2.1 Paradigma Mesin Penerjemah (Bhattacharyya, 2015)

2.2.6 Direct Machine Translation

Penerjemahan sebuah kalimat secara langsung dilakukan dengan cara memproses pemetaan satu per satu kata yang terdapat dalam kalimat dari bahasa sumber menuju bahasa tujuan dengan menggunakan bantuan kamus dwi bahasa. Dalam proses penerjemahan secara langsung, mesin penerjemah tidak mengamati struktur kalimat bahasa sumber melainkan hanya melakukan *pra-pemrosesan* dan analisis morfologi yang dangkal guna menjadikan kalimat tersebut menjadi sebuah daftar kata-kata. Daftar kata-kata yang dihasilkan, dari bahasa sumber, akan dilakukan pencocokan satu per satu dengan menggunakan kamus dwi bahasa. Daftar padanan kata-kata, dari bahasa sumber menuju bahasa target, yang menemui kecocokan dengan kamus dwi bahasa akan dikumpulkan kembali guna dilakukan penyusunan ulang sesuai tata susunan bahasa target. Langkah yang terakhir adalah pembangkitan hasil terjemahan secara langsung secara morfologi

untuk mendapatkan susunan kalimat yang sesuai dengan bahas tujuan (Jurafsky dan Martin, 2009).



Gambar 2.2 *Direct machine translation* (Jurafsky dan Martin, 2009)

2.2.7 BLEU

BLEU (Bilingual Evaluation Understudy) adalah sebuah algoritma yang berfungsi untuk mengevaluasi kualitas dari sebuah hasil terjemahan yang telah diterjemahkan oleh mesin dari satu bahasa alami ke bahasa lain. BLEU mengukur *modified n-gram precision score* antara hasil terjemahan otomatis dengan terjemahan rujukan dan menggunakan konstanta yang dinamakan *brevity penalty*.

Nilai BLEU didapat dari hasil perkalian antara *brevity penalty* dengan rata-rata geometri dari *modified precision score*. Semakin tinggi nilai BLEU, maka semakin akurat dengan rujukan. Nilai dari BLEU berada pada rentang 0 sampai 1. Suatu terjemahan akan mencapai nilai 1 jika terjemahan tersebut identik dengan terjemahan rujukan. Oleh karena itu, meskipun dengan penerjemahan oleh manusia tidak mungkin akan menghasilkan nilai 1. Sangat penting untuk diketahui bahwa semakin banyak terjemahan rujukan per kalimatnya, maka akan semakin tinggi nilainya. Untuk menghasilkan nilai BLEU yang tinggi, panjang kalimat hasil terjemahan harus mendekati panjang dari kalimat referensi dan kalimat hasil terjemahan harus memiliki kata dan urutan yang sama dengan kalimat referensi (Tanuwijaya, 2009). Rumus BLEU dapat dilihat pada gambar 2.3.

$$\begin{aligned}
 BP_{BLEU} &= \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \\
 P_n &= \frac{\sum_{C \in \text{corpus } n\text{-gram} \in C} \sum \text{count}_{clip(n\text{-gram})}}{\sum_{C \in \text{corpus } n\text{-gram} \in C} \sum \text{count}_{(n\text{-gram})}} \\
 BLEU &= BP_{BLEU} \cdot e^{\sum_{n=1}^N w_n \log p_n}
 \end{aligned}$$

Gambar 2.3 Rumus BLEU (*Bilingual Evaluation Understudy*)

Keterangan:

BP = *brevity penalty*

c = jumlah kata dari hasil terjemahan otomatis

r = jumlah kata rujukan

P_n = modified precision score

$w_n = 1/N$ (standar nilai N untuk BLEU adalah 4)

p_n = jumlah n-gram hasil terjemahan yang sesuai dengan rujukan dibagi jumlah n-gram hasil terjemahan

2.2.8 Database

Database atau basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, yang tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan komponen terpenting dalam membangun sebuah sistem informasi, karena digunakan untuk menampung seluruh data yang ada dalam sistem sehingga dapat dieksplorasi. Untuk menyusun informasi - informasi dalam berbagai bentuk. *Database* merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi. Menurut (Stephens, 2000), basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari - hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan data secara terorganisir. Setelah data disimpan, informasi harus mudah diambil. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basis data, dimodifikasi dan dihapus. (Leod, 2001), menyatakan basis data merupakan kumpulan seluruh sumber daya berbasis komputer milik organisasi. Sistem manajemen basis data adalah aplikasi perangkat lunak yang menyimpan struktur basis data, hubungan antar data dalam basis data serta berbagai formulir dan laporan yang berkaitan dengan basis data. Basis data yang dikendalikan oleh sistem manajemen basis data adalah satu set catatan data yang berhubungan dan saling menjelaskan.

Database adalah kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi yang berhubungan atau lebih. Setiap *database* dapat berisi atau memiliki sejumlah objek *database* seperti *field*, *table*, *indeks*, dan lain-lain. Menurut (Connolly dan Begg, 2005), pengertian basis data yaitu kumpulan koleksi data-data yang saling berhubungan secara logika yang isinya didesain untuk memenuhi kebutuhan informasi dari suatu perusahaan.

Ada beberapa istilah umum yang sering dipakai pada *database*, yaitu sebagai berikut :

- a. *Field*, yaitu sekumpulan kecil dari kata atau sebuah deretan angka-angka.
- b. *Record*, yaitu kumpulan dari *field* yang berelasi secara logis.
- c. *File*, yaitu kumpulan dari *record* yang berelasi secara logis.
- d. *Entity*, yaitu orang, tempat, benda, atau kejadian yang berkaitan dengan informasi yang disimpan.
- e. *Attribute*, yaitu setiap karakteristik yang menjelaskan suatu *entity*.
- f. *Primary key*, yaitu sebuah *field* yang nilainya unik yang tidak sama antara satu *record* dengan *record* yang lain.
- g. *Foreign Key*, yaitu sebuah *field* yang nilainya berguna untuk menghubungkan *primary key* yang berbeda pada tabel yang berbeda.

2.2.9 Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi

console, aplikasi *Windows*, ataupun aplikasi *Web*. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*).

Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk platform *Microsoft .NET Framework 3.5*. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform *.NET Framework 2.0* dan *3.0*. Visual Studio 2003 ditujukan untuk *.NET Framework 1.1*, dan Visual Studio 2002 ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan *Microsoft .NET Framework*. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (VS1998).

2.2.10 C#

C# (C sharp) adalah” sebuah bahasa pemrograman berbasis objek yang didukung oleh Microsoft .NET Framework” . Microsoft .NET Framework adalah

perantara agar aplikasi dengan bahasa pemrograman yang didukung dapat berkomunikasi dengan sistem operasi yang digunakan oleh komputer kebanyakan orang. Sama dengan bahasa pemrograman yang lainnya, C# dapat digunakan untuk membangun berbagai macam aplikasi, seperti perangkat lunak berbasis Windows dan perangkat lunak berbasis web serta aplikasi berbasis web services dan juga dapat digunakan untuk *script* pada *Game engine Unity 3D*. C# memiliki berbagai macam fitur seperti penyederhanaan pada pewarisan jamak (*multiple inheritance*), seperti *garbage collection*, *reflection*, akar kelas (*root class*), desain berorientasi objek dan fitur-fitur tersebut di dalam C# kini telah diaplikasikan terhadap *metadata*, *property*, *event*, iterasi, dan konversi antara tipe-tipe sederhana dan objek. Bahasa C# sudah distandarisasi secara internasional oleh ECMA.

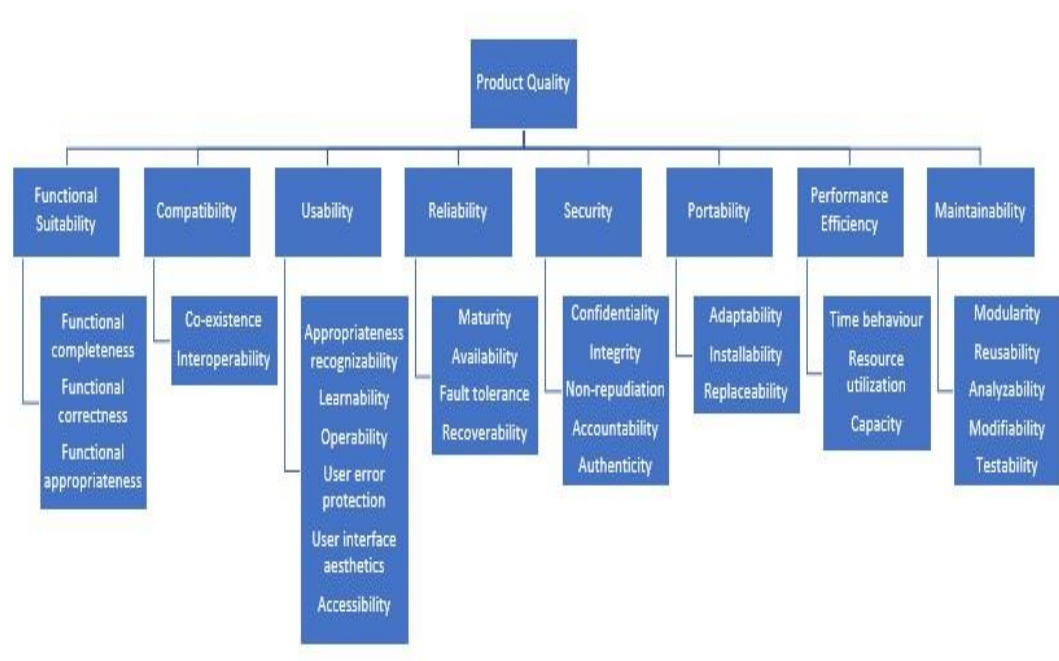
2.2.11 ISO 25010

ISO/IEC 25010 (Babar & Iqbal, 2016) merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang *software engineering*. *Product quality* ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk - perangkat lunak antara lain:

1. Kualitas dalam model penggunaan,
2. Model kualitas produk, dan
3. Data model kualitas

Model kualitas produk terdiri dari delapan karakteristik yang berhubungan dengan sifat statis perangkat lunak dan sifat dinamis dari sistem komputer. Model ini berlaku untuk sistem komputer dan produk perangkat lunak. Karakteristik yang didefinisikan oleh kedua model tersebut relevan untuk semua produk perangkat

lunak dan sistem komputer. Karakteristik dan subkarakteristik memberikan terminologi yang konsisten untuk menentukan, mengukur dan mengevaluasi kualitas sistem dan perangkat lunak. Mereka juga menyediakan seperangkat karakteristik kualitas yang sesuai dengan persyaratan kualitas yang dapat dibandingkan untuk kelengkapan. (ISO/IEC, 2011).



Gambar 2.4 Model Kualitas ISO/IEC 25010

Menurut (Triantoro & Gunawan, 2017) Karakteristik Kualitas ISO/IEC 25010 adalah sebagai berikut:

1. *Functional Suitability*

Sejauh mana perangkat lunak mampu menyediakan fungsi yang memenuhi kebutuhan yang dapat digunakan dalam kondisi tertentu. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu:

- a. *Functional completeness*, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna secara spesifik.

- b. *Functional correctness*, sejauh mana produk atau sistem menyediakan hasil yang benar sesuai kebutuhan.
- c. *Functional appropriateness*, sejauh mana fungsi yang disediakan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.

2. *Compatibility*

Sejauh mana sebuah produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen dan/atau menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat keras dan environment perangkat lunak yang sama. Karakteristik ini dibagi menjadi 2 karakteristik yaitu:

- a. *Co-existence*, sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien sementara berbagi sumber daya dengan produk atau sistem yang lain tanpa merugikan produk atau sistem tersebut.
- b. *Interoperability*, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.

3. *Usability*

Sejauh mana sebuah produk atau sistem dapat digunakan oleh user tertentu untuk mencapai tujuan dengan efektif, *efficiency*, dan kepuasan tertentu dalam konteks penggunaan. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu:

- a. *Appropriateness recognizability*, sejauh mana pengguna dapat mengetahui apakah sistem atau produk sesuai kebutuhan mereka.

- b. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu yang belajar menggunakan sistem atau produk dengan efisien, efektif, kebebasan dari resiko dan kepuasan dalam konteks tertentu.
- c. *Operability*, sejauh mana produk atau sistem mudah dioperasikan dan dikontrol.
- d. *User error protection*, sejauh mana produk atau sistem melindungi pengguna terhadap membuat kesalahan.
- e. *User interface aesthetics*, sejauh mana antarmuka pengguna dari produk atau sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.
- f. *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai tujuan tertentu sesuai konteks penggunaan.

4. *Reliability*

Sejauh mana sebuah sistem, produk atau komponen dapat menjalankan fungsi tertentu dalam kondisi tertentu selama jangka waktu yang ditentukan. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu:

- a. *Maturity*, sejauh mana produk atau sistem mampu memenuhi kebutuhan secara handal di bawah keadaan normal.
- b. *Availability*, sejauh mana produk atau sistem siap beroperasi dan dapat diakses saat perlu digunakan.

- c. *Fault tolerance*, sejauh mana produk atau sistem tetap berjalan sebagaimana yang dimaksud meskipun terjadi kesalahan pada perangkat-keras atau perangkat lunak.
- d. *Recoverability*, sejauh mana produk atau sistem mampu dapat memulihkan data yang terkena dampak secara langsung dan menata ulang kondisi sistem seperti yang diinginkan ketika terjadi gangguan.

5. *Security*

Sejauh mana sebuah produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan level otorisasi yang dimiliki. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu:

- a. *Confidentiality*, sejauh mana produk atau perangkat lunak memastikan data hanya bisa diakses oleh mereka yang berwenang untuk memiliki akses.
- b. *Integrity*, sejauh mana produk atau perangkat lunak mampu mencegah akses yang tidak sah untuk memodifikasi data.
- c. *Non-repudiation*, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan terhadap peristiwa atau tindakan tersebut.
- d. *Accountability*, sejauh mana tindakan dari suatu entitas dapat ditelusuri secara unik untuk entitas.
- e. *Authenticity*, sejauh mana identitas subjek atau sumber daya dapat terbukti menjadi salah satu yang diklaim.

6. *Portability*

Sejauh mana keefektifan dan efisiensi sebuah sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan pada lingkungan yang berbeda. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu:

- a. *Adaptability*, sejauh mana produk atau sistem dapat secara efektif dan efisien disesuaikan pada perangkat lunak, perangkat keras dan lingkungan yang berbeda.
- b. *Installability*, sejauh mana produk atau sistem dapat berhasil dipasang atau dihapus dalam lingkungan tertentu.
- c. *Replaceability*, sejauh mana produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama pada lingkungan yang sama.

7. *Performance Efficiency*

Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu:

- a. *Time behaviour*, sejauh mana respon dan pengolahan waktu produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- b. *Resource utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- c. *Capacity*, sejauh mana batas maksimum parameter produk atau sistem dapat memenuhi persyaratan.

8. *Maintainability*

Sejauh mana keefektifan dan efisiensi dari sebuah produk atau sistem dapat dirawat. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu:

- a. *Modularity*, sejauh mana sistem terdiri dari komponen terpisah sehingga perubahan atau modifikasi pada salah satu komponen tersebut memiliki dampak yang kecil terhadap komponen yang lain.
- b. *Reusability*, sejauh mana aset dapat digunakan lebih oleh satu sistem atau digunakan untuk membangun aset lain.
- c. *Analyzability*, tingkat efektivitas dan efisiensi untuk mengkaji dampak perubahan pada satu atau lebih bagian-bagian produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.
- d. *Modifiability*, sejauh mana produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menurunkan kualitas produk yang ada.
- e. *Testability*, tingkat efektivitas dan efisiensi untuk membentuk kriteria uji dari produk, sistem atau komponen dan uji dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.