



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Game

Game dapat didefinisikan sebagai aktivitas pemecahan masalah, yang didekati dengan sikap bermain-main (Schell, 2008). Fullerton (2014) membagi elemen *game* menjadi dua bagian, yaitu *formal elements* dan *dramatic elements*. *Formal elements* adalah elemen-elemen yang menyusun struktur *game*. Tanpa *formal elements*, sesuatu tidak dapat disebut sebagai sebuah *game*. Terdapat delapan *formal elements* seperti dijabarkan Fullerton, antara lain sebagai berikut.

1. *Players*, yaitu pemain *game*. Elemen ini mencakup jumlah pemain, peran dari tiap pemain, dan interaksi antar pemain.
2. *Objective*, yaitu tujuan yang ingin dicapai pemain dalam batasan *game*.
3. *Procedures*, yaitu metode dan aksi yang dapat dilakukan pemain untuk mencapai *objective*.
4. *Rules*, yaitu definisi objek dan konsep dalam *game*, serta tindakan pemain yang diizinkan. Terdapat beberapa *rules* yang umum antar *game*, diantaranya *rules* yang mendefinisikan objek dan konsep, *rules* yang membatasi tindakan pemain, dan *rules* yang menentukan akibat dari tindakan tertentu.
5. *Resources*, yaitu aset yang dapat digunakan untuk menyelesaikan tujuan tertentu dalam *game*.
6. *Conflict*, yaitu konflik yang muncul akibat pemain yang berusaha mencapai *objective game*, sedangkan terdapat *rules* dan *procedures* yang menghalangi pemain untuk mencapai tujuan tersebut secara langsung.

7. *Boundaries*, yaitu batasan antara *game* dan segala sesuatu yang tidak termasuk didalam *game*.
8. *Outcome*, yaitu hasil akhir dari *game*. *Outcome* dari *game* harus bersifat tidak pasti, untuk menarik minat para pemain.

Sebaliknya, Fullerton (2014) mendefinisikan *dramatic elements* sebagai elemen-elemen yang membuat pemain merasakan pengalaman *game* secara emosional dan mengakibatkan pemain peduli terhadap hasil akhirnya. *Dramatic elements* memberi konteks pada *gameplay*, mengintegrasikan *formal elements* dalam *game* menjadi pengalaman yang berarti bagi pemain. Tidak seperti *formal elements*, *dramatic elements* tidak wajib disertakan semuanya dalam *game*. Terdapat tujuh *dramatic elements* seperti dijabarkan Fullerton, antara lain sebagai berikut.

1. *Challenge*, yaitu tugas-tugas dalam *game* yang terasa memuaskan untuk diselesaikan dan membutuhkan jumlah usaha yang tepat untuk menciptakan perasaan *accomplishment* dan *enjoyment*.
2. *Play*, yaitu kebebasan bergerak dalam kerangka yang lebih kaku. Dalam kasus *game*, kerangka kaku tersebut yaitu *rules* dan *procedures*, sedangkan kebebasan yang dimaksud adalah kebebasan pemain untuk bertindak selama masih mematuhi *rules*.
3. *Premise*, yaitu penetapan *game* didalam sebuah latar atau metafora. Tanpa adanya *dramatic premise*, *game* dapat menjadi terlalu abstrak bagi pemain sehingga pemain tidak dapat terikat secara emosional kepada *game*.
4. *Character*, yaitu agen-agen yang bertindak untuk menceritakan sebuah drama.

5. *Story*, yaitu cerita dalam *game*. Cerita pada *game* berbeda terhadap cerita pada film, televisi, dan pertunjukkan, karena ketidakpastian dalam cerita pada *game* diselesaikan oleh pemain sendiri.
6. *World building*, yaitu desain mendalam dan rumit dari sebuah dunia fiksi. *World building* biasanya mengandung peta dan sejarah, tetapi juga dapat mengandung budaya, bahasa, mitologi, pemerintahan, politik, ekonomi dan lain-lain.
7. *The dramatic arc*, yaitu jumlah ketegangan dramatis dalam cerita seiring berjalannya *game*.

Lingkungan *game* tidak boleh membatasi proses kognitif pemain, sebaliknya lingkungan *game* sebaiknya membiarkan pemain menentukan pilihan secara bebas untuk mencapai tujuan akhir. Lingkungan *game* juga harus menghindari terlalu banyak kebebasan tanpa konsekuensi, karena mengakibatkan *game* menjadi membosankan, dan skenario yang terlalu terkendali, karena membatasi pemikiran pemain (Paras & Bizzocchi, 2005).

2.2 Simulasi

Simulasi yaitu sistem yang merepresentasikan kenyataan yang disederhanakan, dinamis, dan valid. Simulasi berbeda dengan *game* pada modelnya. Model pada sebuah simulasi dinilai berdasarkan perbandingannya dengan kenyataan, dan berdasarkan kecocokan dengan sistem yang direpresentasikan selama bermain. Secara kontras, *game* diciptakan tanpa acuan nyata. Selain itu, *game* mengimplikasikan adanya konflik atau kompetisi yang berusaha

dimenangkan oleh pemain, sedangkan simulasi tidak selalu mengandung hal tersebut (Sauvé dkk., 2010).

2.3 Game Simulasi

Sauvé dkk. (2010) menyatakan *game* simulasi adalah gabungan elemen-elemen *game* dengan karakteristik simulasi. Terdapat minimal seorang pemain yang merupakan bagian dari simulasi, berinteraksi dengan unsur simulasi lainnya, dan memiliki gagasan mengenai menang dan kalah.

Sauvé dkk. (2010) juga mendaftarkan enam atribut-atribut penting dari *game* dalam konteks pembelajaran, antara lain sebagai berikut.

1. Satu atau beberapa pemain. Pemain dapat bermain melawan diri sendiri, dengan orang lain, atau melawan orang lain. Dalam *game* edukasi, pemain juga merupakan pihak yang melakukan pembelajaran, dan yang menerima *feedback* untuk memvalidasi pembelajaran tersebut.
2. Konflik, direpresentasikan dalam *game* sebagai rintangan yang menghalangi tercapainya tujuan *game* secara mudah oleh pemain.
3. Peraturan, yaitu kumpulan kondisi sederhana maupun rumit yang menggambarkan hubungan antara para pemain dan lingkungan *game*.
4. Tujuan yang telah ditentukan oleh *game*, merujuk pada akhir *game*, serta gagasan mengenai kemenangan atau penghargaan. Pada *game* edukasi, tujuan akhir juga mengandung tujuan pembelajaran yang ingin dicapai para pemain.
5. Sifat semu, artinya *game* merupakan aktivitas fiksi tanpa acuan ke dunia nyata, atau beroperasi diluar standar dunia nyata.

6. Sifat *educational*, yaitu *game* harus berkontribusi terhadap pembelajaran, yang didefinisikan sebagai proses mendapatkan pengetahuan akibat interaksi dengan lingkungan.

Sebaliknya, atribut-atribut penting dari simulasi yang berfokus pada pembelajaran antara lain sebagai berikut (Sauvé dkk., 2010).

1. Model dunia nyata yang didefinisikan sebagai sistem. Model dunia nyata yang dideskripsikan dalam simulasi mengandung satu atau lebih elemen dari sistem nyata yang lebih rumit. Penentuan elemen tergantung pada apa yang dititikberatkan oleh perancang simulasi.
2. Model dinamis, artinya simulasi meniru kenyataan ke dalam sebuah model, dan memperbolehkan pemain untuk mengubah model tersebut untuk mempelajarinya sesuai keinginan pemain.
3. Model yang disederhanakan, artinya model tidak sama persis dengan kenyataan, dan mengandung abstraksi yang diperlukan untuk memahami fungsi dan tugas sistem.
4. Model yang akurat dan valid, artinya model harus seragam dan koheren dengan spesifikasi lingkungan, dengan tetap memperhatikan kesesuaian dengan kenyataan. Walaupun telah disederhanakan, simulasi tetap harus akurat karena fungsi utamanya adalah untuk memungkinkan pemahaman yang lebih baik tentang dunia nyata.
5. Sifat *educational*, artinya tujuan simulasi adalah untuk mengembangkan kompetensi, baik sederhana maupun rumit.

Game simulasi menggabungkan atribut-atribut dari *game* dan dari simulasi menjadi tujuh buah atribut, yaitu model dari sistem nyata atau fiksi,

penyederhanaan, dinamis, pemain, konflik, peraturan, dan sifat *educational* yang termasuk dalam tujuan *game* (Sauvé dkk., 2010).

Tabel 2.1 Atribut *Game*, Simulasi, dan *Game Simulasi* (Sauvé dkk., 2010)

<i>Game</i>	Simulasi	<i>Game Simulasi</i>
Bersifat semu	Kenyataan didefinisikan sebagai sistem	Kenyataan didefinisikan sebagai sistem
	<ul style="list-style-type: none"> • Model • Penyederhanaan • Dinamis • Akurat dan valid 	<ul style="list-style-type: none"> • Model • Penyederhanaan • Dinamis
<ul style="list-style-type: none"> • Pemain • Konflik • Peraturan • Tujuan yang telah ditentukan 		<ul style="list-style-type: none"> • Pemain • Konflik • Peraturan • Tujuan yang telah ditentukan

2.4 Mekanika Klasik

Mekanika klasik yaitu ilmu yang mempelajari tentang pergerakan benda, sesuai dengan prinsip-prinsip umum yang dijabarkan oleh Isaac Newton. Mekanika klasik memiliki banyak penerapan penting dalam cabang sains lainnya, di antaranya Astronomi, Kimia, Geologi, dan lain-lain (Fitzpatrick, 2006).

Mekanika klasik atau mekanika Newton menganalisis gerak suatu objek berdasarkan gaya yang dialami objek tersebut dari lingkungannya (French, 1971), tetapi tidak berlaku untuk fenomena dalam skala sangat kecil, seperti getaran dan perputaran molekul, elektron di dalam atom, dan fenomena nuklir atau lebih kecil dari nuklir (Bettini, 2016).

Menurut Giancoli (2016), mekanika dibagi menjadi dua bagian, yaitu kinematika dan dinamika. Kinematika menjelaskan bagaimana benda bergerak. Pokok bahasan kinematika di antaranya perpindahan (*displacement*), kecepatan rata-rata (*average velocity*), kecepatan instan (*instantaneous velocity*), percepatan (*acceleration*), dan kecepatan relatif (*relative velocity*).

Sebaliknya, Giancoli (2016) mendefinisikan dinamika sebagai ilmu yang membahas hubungan antara gaya dan pergerakan. Pokok bahasan dinamika diantaranya massa (*mass*), berat (*weight*), gaya (*force*), dan ketiga hukum Newton tentang pergerakan (*Newton's law of motion*).

2.5 Virtual Reality

Virtual Reality dapat didefinisikan sebagai penggunaan komputer dan teknologi terkait untuk membuat sebuah lingkungan buatan yang mensimulasikan sebuah lingkungan target (Connolly, 2005). Terdapat empat komponen utama pada program *virtual reality* (Senovsky & Kodym, 1999), yaitu sebagai berikut.

1. Semua hal terjadi dalam waktu yang sebenarnya (*real time*).
2. Lingkungan buatan bersifat tiga dimensi.
3. Pengguna dapat memasuki lingkungan tersebut dan bergerak bebas di dalamnya.
4. Lingkungan tersebut tidak bersifat statis dan bagian-bagiannya dapat dimanipulasi pengguna.

Pantelidis (1997) telah merumuskan langkah-langkah untuk mengaplikasikan *virtual reality* pada pelajaran tertentu, yaitu sebagai berikut.

1. Tujuan-tujuan dari pelajaran didefinisikan.

2. Tujuan-tujuan yang dapat menggunakan simulasi sebagai alat ukur atau cara ditandai.
3. Tiap tujuan yang telah ditandai diuji, apakah bisa menggunakan simulasi komputer untuk pencapaiannya atau pengukuran, dan apakah bisa menggunakan simulasi interaktif tiga dimensi.
4. Ditentukan tingkat realisme yang diperlukan untuk tiap tujuan yang lulus uji, dari sangat simbolik hingga sangat nyata.
5. Ditentukan tipe interaksi yang diperlukan dalam skala dari tidak ada interaksi (seperti *desktop VR*) hingga interaksi penuh (seperti menggunakan *head-mounted display*).
6. Ditetapkan tipe *output* indrawi yang diinginkan dari dunia *virtual*, contohnya *haptic* (sentuhan), suara tiga dimensi, atau hanya visual.
7. *Software* dan *hardware virtual reality* dipilih.
8. Lingkungan *virtual* didesain dan dibangun, baik oleh guru, murid ataupun keduanya.
9. Hasil lingkungan *virtual* dievaluasi dengan sebuah *pilot* atau grup murid.
10. Lingkungan *virtual* dimodifikasi sesuai hasil evaluasi.

Modifikasi dilakukan terus hingga lingkungan *virtual* secara sukses mengukur atau membantu meraih tujuan pelajaran.

2.6 Perlin Noise

Perlin Noise adalah algoritma berupa fungsi yang menerima sebuah vektor tiga dimensi sebagai argumennya, dan mengembalikan sebuah nilai skalar. Untuk dua argumen yang berdekatan (dua titik yang berdekatan pada ruang tiga dimensi),

fungsi ini akan mengembalikan nilai skalar yang berdekatan. Fungsi *Noise()* pertama kali dikembangkan oleh Ken Perlin dan dijabarkan pada *paper* berjudul *An image Synthesizer* pada 1985. Prosedur dari fungsi *Noise()* pada *paper* tersebut yaitu sebagai berikut (Perlin, 1985).

1. Ditentukan sebuah himpunan dari semua titik di ruang tiga dimensi, dimana koordinat x , y , dan z dari titik-titik tersebut merupakan bilangan bulat. Himpunan ini disebut *integer lattice*.
2. Sebuah nilai *pseudo-random* dan nilai gradien untuk x , y , dan z diasosiasikan untuk tiap titik dalam *integer lattice*. Dengan kata lain, untuk setiap bilangan bulat x , y , dan z ditentukan 4 bilangan *pseudo-random* a , b , c , dan d , dimana $[a, b, c]$ adalah gradien pada ketiga dimensi dan d merupakan nilai *Noise()* pada titik x , y , z .

$$H(x, y, z) = [a, b, c, d] \quad \dots(2.1)$$

(x, y, z) merupakan koordinat titik pada sumbu x , y , dan z ; $[a, b, c, d]$ merupakan persamaan linear dengan gradien $[a, b, c]$ dan nilai d ; dan $H()$ merupakan fungsi *hash*.

3. Bila argumen x , y , dan z dari fungsi *Noise*(x , y , z) merupakan bilangan bulat, maka:

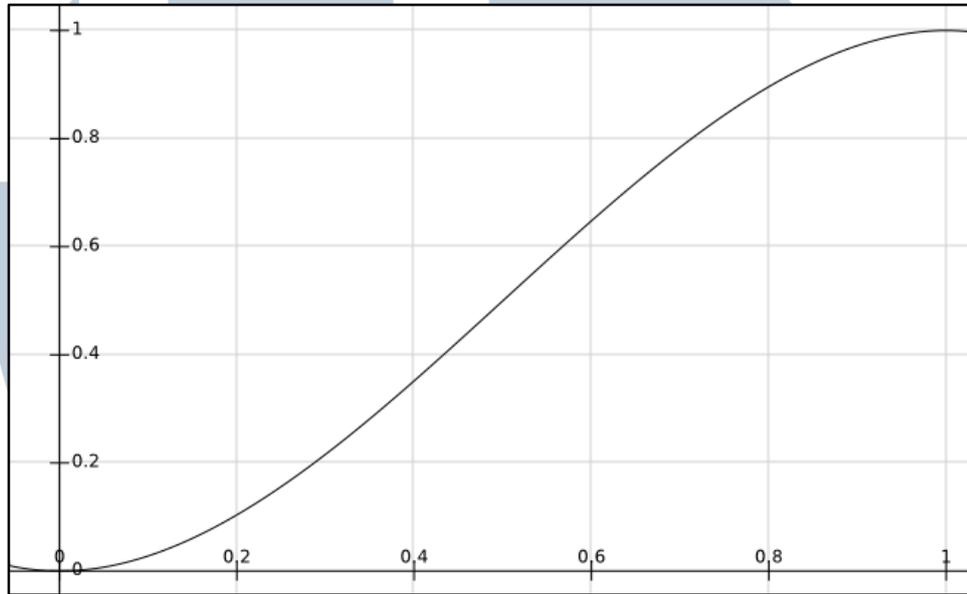
$$Noise(x, y, z) = D_{[x,y,z]} \quad \dots(2.2)$$

$D_{[x,y,z]}$ merupakan nilai d seperti ditentukan pada langkah 2.

4. Selain itu maka dikalkulasikan interpolasi antar nilai gradien x , y , dan z pada delapan titik terdekat dalam *integer lattice* untuk menjadi nilai $[a, b, c]$ dengan fungsi $s(t)$ seperti pada Rumus (2.3), diaplikasikan pada ketiga gradien.

$$s(t) = 3t^2 - 2t^3 \quad \dots(2.3)$$

t adalah nilai gradien a, b, c pada *integer lattice* terdekat, dan s adalah fungsi interpolasi.



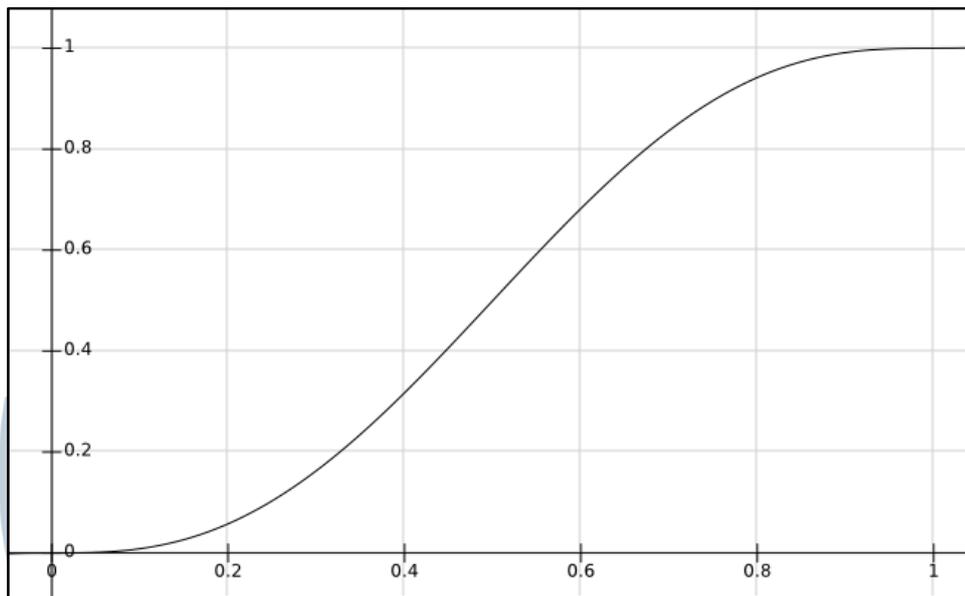
Gambar 2.1 Fungsi Interpolasi $s(t)$ (Perlin, 1985)

Perlin menunjukkan penerapan *Noise* dalam beberapa kasus, di antaranya untuk membuat tekstur permukaan yang tampak *random*, menyimulasikan api, awan dan permukaan air secara prosedural (Perlin, 1985).

Perlin kemudian mengubah algoritma *Perlin Noise* untuk memperbaiki dua hal, yaitu diskontinuitas interpolasi pada turunan kedua $s(t)$, dan komputasi gradien yang kurang optimal. Perubahan yang dilakukan yaitu mengubah rumus $s(t)$ menjadi Rumus (2.4).

$$s(t) = 6t^5 - 15t^4 + 10t^3 \quad \dots(2.4)$$

t adalah nilai gradien a, b, c pada *integer lattice* terdekat, dan s adalah fungsi interpolasi.



Gambar 2.2 Fungsi Interpolasi $s(t)$ pada *Improving Noise* (Perlin, 2002)

Selain itu ketiga gradien x , y , dan z juga diubah menjadi salah satu antara $(1, 1, 0)$, $(-1, 1, 0)$, $(1, -1, 0)$, $(-1, -1, 0)$, $(1, 0, 1)$, $(-1, 0, 1)$, $(1, 0, -1)$, $(-1, 0, -1)$, $(0, 1, 1)$, $(0, -1, 1)$, $(0, 1, -1)$, $(0, -1, -1)$. Gradien untuk tiap *integer lattice* dipilih secara acak dari keduabelas vektor tersebut (Perlin, 2002).

Pada penelitian Pi dkk. (2006), *Perlin Noise* digunakan untuk memberikan detail geometri pada *terrain*. Setelah menciptakan nilai *noise* pada titik-titik diskrit, nilai *noise* pada titik-titik lainnya dapat dihitung melalui interpolasi. Hasil *Perlin Noise* digunakan sebagai *heightmap* untuk *terrain* (Pi dkk., 2006).

2.7 Unity

Unity adalah sebuah *game engine cross-platform* dan *development environment* yang bersifat *user friendly* (Zamojc, 2012). Unity memiliki misi untuk menciptakan alat yang mempermudah *developer* kecil dan independen dalam membuat *game* 3D berkualitas tinggi dan dapat dijalankan pada berbagai *platform* (Takahashi, 2014).

Menurut Patil dan Alvares (2015), fungsi dari Unity sangat banyak. Semua proses pengembangan *game* mungkin dilakukan dengan Unity, diantaranya *shader*, *physics engine*, *network*, *terrain manipulation*, *audio*, *video*, dan animasi.

Unity menggunakan *physics engine* NVIDIA PhysX yang mendukung simulasi pakaian, interaksi *soft* dan *rigid-body*, *ragdoll*, sistem sendi, dan sistem tumbukan. Unity juga mendukung pembuatan *terrain* dengan *terrain painting*, *detail texture painting*, dan *tree creator* (Patil & Alvares, 2015).

2.8 Skala Likert

Skala Likert adalah sekumpulan pernyataan yang ditawarkan untuk sebuah situasi nyata maupun hipotetis. Partisipan kemudian diminta untuk menunjukkan tingkat persetujuan (dari sangat tidak setuju hingga sangat setuju) terhadap pernyataan-pernyataan tersebut dalam skala metrik (Joshi dkk., 2015).

Menurut Sugiyono (2014), dengan skala Likert, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pernyataan atau pertanyaan.

Sugiyono (2014) juga memberi contoh analisis hasil data angket dengan skala Likert. Jawaban diberi skor antara lain sebagai berikut.

1. Sangat setuju / selalu / sangat positif diberi skor 5.
2. Setuju / sering / positif diberi skor 4.
3. Ragu-ragu / kadang-kadang / netral diberi skor 3.
4. Tidak setuju / hampir tidak pernah / negatif diberi skor 2.
5. Sangat tidak setuju / tidak pernah / sangat negatif diberi skor 1.

Kemudian setelah diberikan kepada 100 orang, hasil data angket misalnya sebagai berikut (Sugiyono, 2014).

1. 25 orang menjawab sangat setuju.
2. 40 orang menjawab setuju.
3. 5 orang menjawab ragu-ragu.
4. 20 orang menjawab tidak setuju.
5. 10 orang menjawab sangat tidak setuju.

Data interval tersebut kemudian dianalisis dengan menghitung rata-rata jawaban berdasarkan skor setiap jawaban dari responden (Sugiyono, 2014).

Tabel 2.2 Contoh Analisis Data Angket (Sugiyono, 2014)

Jumlah skor untuk 25 orang yang menjawab sangat setuju	25 x 5	125
Jumlah skor untuk 40 orang yang menjawab setuju	40 x 4	160
Jumlah skor untuk 5 orang yang menjawab ragu-ragu	5 x 3	15
Jumlah skor untuk 20 orang yang menjawab tidak setuju	20 x 2	20
Jumlah skor untuk 10 orang yang menjawab sangat tidak setuju	10 x 1	10
Jumlah total		350

Jumlah skor ideal untuk seluruh *item* = $5 \times 100 = 500$. Jumlah skor yang diperoleh dari penelitian = 350. Berdasarkan data tersebut maka tingkat persetujuan responden yaitu $(350 / 500) \times 100\% = 70\%$ dari yang diharapkan (Sugiyono, 2014).

2.9 Usability

Menurut Nielsen (2012), *usability* adalah *quality attribute* yang menguji seberapa mudah suatu *user interface* digunakan. Brooke (1986) menyatakan bahwa *usability* bukanlah kualitas yang nyata atau absolut. Brooke (1986) juga menyatakan bahwa *usability* dapat dijabarkan sebagai kualitas umum dari kecocokan sesuatu terhadap sebuah tujuan, dan *usability* dari alat atau sistem apapun harus dilihat dari konteks di mana alat atau sistem tersebut digunakan.

2.10 System Usability Scale

System Usability Scale (SUS) adalah skala sederhana yang terdiri dari sepuluh *item* untuk memberi pandangan global terhadap penilaian subjektif untuk *usability*. Sepuluh *item* tersebut antara lain sebagai berikut (Brooke, 1986).

1. *I think that I would like to use this system frequently.*
2. *I found the system unnecessarily complex.*
3. *I thought the system was easy to use.*
4. *I think that I would need the support of a technical person to be able to use this system.*
5. *I found the various functions in this system were well integrated.*
6. *I thought there was too much inconsistency in this system.*
7. *I would imagine that most people would learn to use this system very quickly.*
8. *I found the system very cumbersome to use.*
9. *I felt very confident using the system.*
10. *I needed to learn a lot of things before I could get going with this system.*

SUS menggunakan skala Likert lima poin, mulai dari “*strongly agree*” hingga “*strongly disagree*”. Hasil SUS berupa sebuah angka yang merepresentasikan pengukuran gabungan dari *overall usability* sistem (Brooke, 1986).

Bangor dkk. (2008) menyatakan bahwa SUS merupakan pilihan baik untuk mengukur *usability* secara umum, antara lain karena bersifat *technology agnostic*; relatif lebih cepat dan mudah untuk partisipan survei; menyediakan sebuah nilai yang mudah dimengerti oleh banyak orang, dari *project manager* hingga *programmer*; dan tidak dipatenkan, sehingga bersifat *cost-effective*.

Menurut Brooke (1986), untuk mengukur nilai SUS, pertama jumlahkan semua kontribusi nilai per *item*. Kontribusi nilai per *item* antara 0 hingga 4. Untuk *item* nomor 1, 3, 5, 7, dan 9, kontribusi nilai yaitu posisi skala dikurangi satu. Untuk *item* nomor 2, 4, 6, 8, dan 10, kontribusi nilai yaitu 5 dikurangi posisi skala. Jumlah semua nilai dikali dengan 2,5 adalah nilai *usability* sistem. Nilai SUS bervariasi antara 0 dan 100. Sauro (2011) menyatakan bahwa nilai SUS di atas 68 dapat dianggap di atas rata-rata, dan nilai di bawah 68 dapat dianggap di bawah rata-rata.

Santoso dan Sharfina (2016) menerjemahkan dan mengadaptasi SUS menjadi versi bahasa Indonesia. Hasil terjemahan SUS antara lain sebagai berikut.

1. Saya berpikir akan menggunakan sistem ini lagi.
2. Saya merasa sistem ini rumit untuk digunakan.
3. Saya merasa sistem ini mudah untuk digunakan.
4. Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5. Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6. Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini,.
7. Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8. Saya merasa sistem ini membingungkan.
9. Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10. Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.