

## **BAB II**

### **TINJAUAN PUSTAKA**

Berikut diuraikan beberapa teori dan penelitian terdahulu yang relevan dengan permasalahan yang diangkat, diharapkan dapat menjadi landasan ilmiah tentang perumusan metode dan pemecahan masalah.

Penelitian yang dilakukan oleh (Agung Ramadhanu, Rahmatul Husna Arsyah, Neni Sri Wahyuni Nengsi, Nurhaliza, Januari 2019) yang berjudul “Rancang Bangun Aplikasi *Mobile* Edukasi Mengenai *Parenting Skills* Bagi Orang Tua Berbasis Android Dengan Menggunakan Bahasa Pemrograman Java Android” menerangkan bahwa terdapat satu alternatif untuk dapat memberikan pemahaman bagi orang tua dalam mendidik anak-nya, yaitu sebuah aplikasi *parenting skills* berbasis android. Dari hasil perancangan dan implementasi yang dilakukan, Aplikasi *Parenting Skills* ini dapat dirancang dengan baik, sehingga berguna bagi orang tua untuk menambah pemahaman dalam mendidik anak. Aplikasi *Parenting Skills* ini dapat diakses di android sehingga para orang tua tidak kesulitan dalam menambah pemahaman tentang *parenting skills* tersebut.

Di dalam jurnal penelitian yang ditulis oleh (Elly Mufida, Martini, Ady Hermawan, Mei 2018) dengan judul “Perancangan Aplikasi *Parenting* Penguatan Perilaku Positif Anak Oleh Orang Tua Berbasis Android” menjelaskan bahwa, saat ini diperlukan sebuah aplikasi *mobile* untuk orang tua dan anak yang dapat membantu mengurangi perilaku negatif anak menjadi ke arah yang lebih positif. Metode *Waterfall* digunakan dalam penelitian dan pembangunan aplikasi *parenting* ini. Sedangkan untuk pembuatan aplikasi penulis memilih Android Studio sebagai *Integrated Development Environment* (IDE) untuk pengembangan aplikasi.

Aplikasi *Parenting* berbasis Android ini dapat membantu orang tua dalam pembentukan perilaku positif anak disamping kesibukannya.

Penelitian yang dilakukan oleh (Jansen Wiratama, Hari Santoso, November 2019) dengan judul “Perancangan Sistem Aplikasi Berbasis *Mobile* Untuk Mengurangi Tindakan Ancaman Atau Kekerasan Pada Pelajar (Implementasi: Sekolah Dharma Putra)” menyebutkan bahwa kekerasan pada pelajar masih menjadi salah satu masalah yang dihadapi oleh para pelajar, orang tua dan pihak sekolah. Kurangnya keterbukaan dari pelajar dan keterbatasan orang tua serta pihak sekolah dalam melakukan pemantauan aktifitas pelajar mengakibatkan tindak kekerasan dan ancaman pada pelajar. Diperlukan sebuah sistem aplikasi yang membuat para pelajar terhubung dengan orang tuanya masing-masing melalui *smartphone* pribadi para pelajar dan orang tua sehingga diharapkan dapat mengurangi tingkat kekerasan khususnya di kalangan pelajar. Sistem aplikasi yang akan dirancang berbasis *mobile* dan pada tahap perancangannya akan menggunakan metode RAD (*Rapid Application Development*).

Dalam penelitian yang dilakukan oleh (Agitya, 2020) dengan judul “Pembangunan Aplikasi *Parental Supervision* Untuk Pengawasan Orangtua Terhadap Anak Berbasis Android” menyebutkan kasus kenakalan remaja semakin tinggi, yang membuat orang tua perlu berusaha lebih dalam mengawasi anak mereka. Berdasarkan hal diatas, dapat disimpulkan perlunya membangun perangkat lunak yang dapat membantu permasalahan para orang tua. *Parental Supervision* adalah sebuah perangkat lunak yang berjalan pada perangkat *mobile* sehingga mudah untuk diakses dan di bawa kemana saja. Perangkat lunak yang dibangun menggunakan pemodelan berbasis objek. Berdasarkan hasil pengujian

menggunakan metode pengujian *alpha* dan *beta* dapat disimpulkan perangkat lunak *Parental Supervision* telah membantu orang tua dalam mengawasi anak dengan menggunakan *platform* Android walaupun disela-sela kesibukan orangtua.

Dalam jurnal yang ditulis oleh (Pramita Puspaningtyas, Budi Utomo, Kemal Nazarudin, 2022) dengan judul “Model Konsep Perancangan Sistem Informasi Mobile *Parenting* untuk Pemantauan Tumbuh Kembang Balita di *Daycare*”, yang bertujuan untuk mengembangkan model konsep perancangan sistem informasi tumbuh kembang anak berbasis *less paper* untuk membuat pencatatan, pelaporan, serta pemantauan kesehatan balita efektif dan efisien. Metode yang digunakan adalah *literature review* dan *system development*. Sistem dapat digunakan oleh orang tua dalam memantau tumbuh kembang anak secara *mobile*, dan juga dapat digunakan oleh pengasuh *daycare* dalam pencatatan hasil pengkajian secara langsung ke dalam aplikasi.

## **2.1 Parenting**

Penelitian yang dilakukan oleh (Agung Ramadhanu, Rahmatul Husna Arsyah, Neni Sri Wahyuni Nengsi, Nurhaliza, Januari 2019) menjelaskan bahwa *parenting* merupakan proses interaksi berkelanjutan antara orang tua dan anak-anak mereka yang meliputi aktivitas-aktivitas seperti memberi makan, memberi petunjuk dan melindungi anak-anak ketika mereka tumbuh. Aktivitas *parenting* biasanya terjadi dalam lingkungan keluarga, namun *parenting* tidak terbatas hanya pada mereka yang melahirkan anak. Tanggung jawab *parenting* juga dilakukan oleh pihak-pihak lain dalam masyarakat, seperti para guru di sekolah, pembantu rumah tangga, perawat bayi (*Baby Sitter*), serta media masa (Televisi, Majalah dan Surat Kabar). Kendati demikian, orang tua adalah pihak yang paling bertanggung jawab

dalam mengasihani dan memperhatikan anak-anak serta menolong mereka tumbuh. *Parenting Skill* merupakan suatu keterampilan orang tua dalam mendidik atau mengasuh anaknya agar menjadi pribadi yang lebih baik.

Beberapa definisi tentang pengasuhan menunjukkan bahwa konsep pengasuhan mencakup beberapa pengertian pokok, antara lain:

1. Pengasuhan bertujuan untuk mendorong pertumbuhan dan perkembangan anak secara optimal, baik secara fisik, mental maupun sosial.
2. Pengasuhan merupakan sebuah proses interaksi yang terus menerus antara orang tua dengan anak.
3. Pengasuhan adalah sebuah proses sosialisasi. Sebagai sebuah proses interaksi dan sosialisasi, proses pengasuhan tidak bisa dilepaskan dari sosial budaya dimana anak dibesarkan.

### **2.1.1 Seni Bernegosiasi Dengan Anak**

Buku karya (Yoga, 2021) yang berjudul “Seni Bernegosiasi Dengan Anak” menjelaskan, negosiasi antara orang tua dan anak merupakan pengalaman penting bagi mereka karena dapat mengajarkan anak untuk menghadapi masalah dengan baik. Negosiasi tidak berarti orang tua menyerah pada kemauan anak. Negosiasi berarti anak dan orang tua sama-sama memiliki hak untuk menyuarakan keinginan, sekaligus kewajiban untuk mendengarkan. Bernegosiasi dengan anak juga tidak perlu melibatkan segala macam teriakan yang hanya akan menimbulkan perdebatan sengit. Dalam buku ini (Yoga, 2021), para orang tua akan belajar belajar bagaimana “menaklukan” buah hati agar mereka mau diajak bicara, mendengarkan, mengikuti arahan, dan memahami tentang apa yang orang tua inginkan. Selain itu, dipaparkan juga berbagai tips dan trik untuk bernegosiasi dengan anak, seperti membangun

budaya curhat, seni afirmasi, keajaiban pujian, sampai bagaimana manajemen konflik yang baik. Semuanya tanpa perlu menaikkan volume suara dan ancaman.

### **2.1.2 Ayah Edy Menjawab 100 Persoalan Sehari-hari Orang Tua**

Siapa yang tidak mengenal sosok Ayah Edy? Penggagas program “*Indonesian Strong From Home*” tersebut merupakan seorang praktisi *parenting* yang telah menggeluti dunia *parenting* dan pendidikan di Indonesia selama lebih dari 14 tahun. Banyak buku yang telah diterbitkan oleh Ayah Edy, salah satunya adalah buku yang berjudul “Ayah Edy Menjawab 100 Persoalan Sehari-hari Orang Tua Yang Tidak Ada Jawabannya Di Kamus Manapun”. Buku ini salah satu buku *parenting best seller* dengan rating pembaca 4.5 dari 5. Buku ini membahas 100 pertanyaan mengenai permasalahan sehari-hari yang sering dialami orang tua. Tidak hanya berisi beragam pertanyaan seputar *parenting*, buku ini juga memberikan solusi dan cara menyelesaikan permasalahan terkait.

“Kenapa ya anak saya suka berbohong?”

“Kapan waktu yang tepat untuk mengajarkan disiplin pada anak?”

“Bagaimana caranya Si Kakak tidak iri pada adik barunya?”

“Bagaimana mengatasi anak yang super aktif atau tidak bisa diam?”

“Bagaimana mengatasi anak yang menjadikan tangisan sebagai senjata untuk mendapatkan keinginannya?”

Pertanyaan tersebut diatas pasti pernah terlintas dan cukup membuat pusing hingga kewalahan mengatasinya. Dalam buku ini (Edy, 2012), pertanyaan tersebut langsung dijawab oleh pakar dengan lugas, tepat, dan dengan gaya bahasa yang ringan dan mudah dimengerti.

### 2.1.3 *The Danish Way of Parenting*

Buku (Jessica Joelle Alexander, Iben Dissing Sandahl, 2016) ini ditulis oleh Jessica Joelle Alexander yang merupakan seorang penulis, kolumnis, dan peneliti bidang budaya berkebangsaan Amerika. Ia meraih gelar di bidang psikologi dan mengajar komunikasi dan keterampilan menulis di Skandinavia dan Eropa Tengah. Sedangkan Iben Dissing Sandahl merupakan pelatih tersertifikasi, penulis, dan psikoterapis naratif berlisensi, MPF yang membuka praktik di luar Kopenhagen, Denmark. Konseling keluarga dan anak adalah spesialisasinya.

Selama lebih dari 40 tahun Denmark selalu terpilih menjadi negara paling bahagia sedunia, menurut World Happiness Report oleh PBB. Tak terhitung banyaknya artikel dan kajian yang berusaha memecahkan misteri ini. Setelah riset bertahun-tahun, ternyata jawabannya sangat sederhana. Ini karena gaya pengasuhan mereka. Filosofi orang Denmark dalam membesarkan anak terbukti memberikan hasil yang cukup efektif, anak-anak yang tangguh, emosi terkendali, dan bahagia. Warisan inilah yang membuat Denmark selalu menempati urutan pertama indeks kebahagiaan seluruh dunia.

## 2.2 **Android**

Menurut (F. Sifauttijani, T. Listyorini dan R. Meimaharani, April 2017) Android merupakan sebuah sistem operasi modifikasi dari Linux yang digunakan untuk *Smartphone* dan *Tablet* hingga perangkat jam tangan sampai televisi pintar. Salah satu sistem operasi yang dapat dioperasikan pada *smartphone* adalah operasi Android yang bersifat *open source*. Android memiliki kekurangan dan kelebihan yaitu:

1. Android sangat mendukung *multitasking* pada aplikasi.
2. Peningkatan kemampuan *copy-paste*.
3. Penggunaan *Browser Chrome* yang lebih cepat ketika diakses menggunakan internet. Selain itu pengguna juga dapat melakukan sinkronisasi dengan *Browser Chrome* yang ada di komputer.
4. *Drag and Drop* memudahkan operasional file dan data. Dan Notifikasi yang sangat mudah.

### 2.3 Android Studio

Android Studio merupakan salah satu *tools* yang dapat digunakan untuk membuat aplikasi berbasis *mobile*. Dalam buku karya (A. Satyaputra dan M.Aritonang, 2016) Android Studio adalah lingkungan pengembangan android baru berdasarkan IntelliJ IDEA, yang menyediakan alat pengembangan android yang terintegrasi dalam pengembangan dan *debugging* program. IntelliJ IDEA sendiri merupakan sebuah komersial Java IDE *powerfull*. IDEA mendukung bahasa seperti Java.

### 2.4 Firebase

Firebase adalah suatu layanan dari Google yang memberikan kemudahan untuk para *Developer* aplikasi dalam mengembangkan aplikasinya. Firebase atau sering dikenal BaaS (*Backend as a Service*) merupakan solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan *Developer* (Intern, Apa itu Firebase? Pengertian, Jenis-jenis, dan Fungsi Kegunaannya, 2020).

Firebase didirikan pertama kali pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk Firebase yang pertama kali adalah Realtime Database. Realtime Database digunakan *Developer* untuk menyimpan data

dan *synchronize* ke banyak *user*. Kemudian berkembang sebagai layanan pengembang aplikasi pada bulan Oktober 2014, perusahaan tersebut diakuisisi oleh Google (Intern, Apa itu Firebase? Pengertian, Jenis-jenis, dan Fungsi Kegunaannya, 2020). Layanan-layanan yang tersedia dari Firebase ada 2 pilihan, di antaranya:

1. SPARK : *Developer* dapat menggunakan layanan secara gratis.
2. BLAZE : *Developer* akan dikenakan biaya sesuai dengan pemakaian layanan.

## 2.5 Figma

Figma adalah aplikasi desain berbasis *cloud* dan alat *prototyping* untuk projek digital. Aplikasi ini beroperasi pada sistem desktop untuk Mac OS dan Windows. Figma berfokus pada penggunaan antarmuka pengguna dan kolaborasi waktu secara nyata (*real-time*) (Kadas, 2022).

## 2.6 Unified Modelling Language (UML)

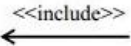
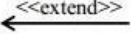
Dalam pembuatan sebuah aplikasi dibutuhkan sebuah pemodelan. Pemodelan sendiri merupakan bentuk implementasi sistem bagaimana meletakkan suatu rancangan ke dalam sebuah gambar (*visual*) yang berbentuk diagram. UML memberikan kemudahan dalam menganalisa, memahami, dan mempermudah pembuatan sebuah program atau sistem. Dilansir dari (Intern, Apa itu UML?, 2021) UML merupakan suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk faktor-faktor *scalability*, *robustness*, *security*, dan sebagainya.



### 2.6.1 Use Case Diagram

Langkah awal untuk melakukan pemodelan adalah perlu adanya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada *Use Case Diagram*. *Use Case Diagram* adalah satu jenis dari diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. (Intern, Apa itu UML?, 2021).

Secara umum *Use Case Diagram* merupakan pola perilaku sistem dan urutan transaksi yang berhubungan yang dilakukan oleh satu aktor. Gambar 2.1 menunjukkan komponen dari *Use Case Diagram*.

| Simbol                                                                              | Keterangan                                                                                                                      |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|   | Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>                           |
|  | <i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor                                                               |
|  | <i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>                                              |
|  | <i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>                          |
|  | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya                        |
|  | Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi |

Gambar 2. 1 Komponen *Use Case Diagram*  
Sumber : (Mery, 2021)

### 2.6.2 Activity Diagram

Dilansir dari (Rizky, 2019) *Activity Diagram* merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. *Activity Diagram* juga digunakan untuk mendefinisikan atau mengelompokkan aliran tampilan dari sistem tersebut. *Activity Diagram* memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Panah tersebut mengarah ke-urutan aktivitas yang terjadi dari awal hingga akhir. *Activity Diagram* menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *Use Case* pada *Use Case Diagram*. Gambar 2.2 menunjukkan komponen dari *Activity Diagram*.

| Simbol                                                                              | Nama                   | Keterangan                                                                                   |
|-------------------------------------------------------------------------------------|------------------------|----------------------------------------------------------------------------------------------|
|   | Status awal            | Sebuah diagram aktivitas memiliki sebuah status awal.                                        |
|  | Aktivitas              | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.               |
|  | Percabangan / Decision | Percabangan dimana ada pilihan aktivitas yang lebih dari satu.                               |
|  | Penggabungan / Join    | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.          |
|  | Status Akhir           | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir    |
|  | Swimlane               | Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

Gambar 2. 2 Komponen *Activity Diagram*  
Sumber : (Juliarto, 2021)

### 2.6.3 Class Diagram

Dilansir dari (Setiawan, Dicoding, 2021) *Class Diagram* merupakan salah satu jenis diagram struktur pada UML (*Unified Modelling Language*) yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. *Class Diagram* bersifat statis, *Class Diagram* tidak menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi. *Class Diagram* memiliki fungsi untuk menunjukkan struktur dari suatu sistem dengan jelas. *Class Diagram* memiliki 3 komponen penyusun. Gambar 2.3 menunjukkan komponen penyusun dari *Class Diagram*.



Gambar 2. 3 Komponen *Class Diagram*  
Sumber : (Setiawan, Dicoding, 2021)

#### 1. Komponen Atas

Komponen ini berisikan nama *Class*. Setiap *Class* memiliki nama yang berbeda-beda, sebutan lain untuk nama ini adalah *simple name* (nama sederhana).

#### 2. Komponen Tengah

Komponen ini berisikan atribut dari *Class*. Komponen ini digunakan untuk menjelaskan kualitas dari suatu *Class*. Atribut ini dapat menjelaskan dapat ditulis

lebih detail, dengan cara memasukan tipe nilai. Atribut dapat memiliki salah satu sifat berikut:

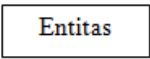
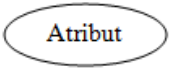
- a) *Private* : Tidak dapat dipanggil dari luar *Class* yang bersangkutan.
- b) *Protected* : Hanya dapat dipanggil oleh *Class* yang bersangkutan dan anak-anak yang mewarisinya.
- c) *Public* : Dapat dipanggil oleh siapa saja.

### 3. Komponen Bawah

Komponen ini menyertakan operasi yang ditampilkan dalam bentuk daftar. Operasi ini dapat menggambarkan bagaimana suatu *Class* dapat berinteraksi dengan data.

#### 2.7 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* atau biasa disingkat ERD merupakan suatu bentuk diagram yang menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi. ERD digunakan untuk menyusun struktur data dan hubungan antar data yang digambarkan dengan menggunakan notasi, simbol, bagan dan lain sebagainya (Baharsyah, 2020). Gambar 2.4 menunjukkan komponen yang terdapat di dalam ERD.

| Notasi                                                                              | Keterangan                                                                                                         |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|  | Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.                                     |
|  | Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.                                        |
|  | Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah). |
|  | Garis sebagai penghubung antara relasi dan entitas atau relasi dan entitas dengan atribut.                         |

Gambar 2. 4 Komponen *Entity Relationship Diagram* (ERD)

Sumber : (Exa, 2022)

## 2.8 Metode Pengembangan Perangkat Lunak

Model pengembangan perangkat lunak yang digunakan oleh penulis dalam membangun aplikasi ini adalah Model *Waterfall*, karena model ini memberikan pendekatan secara terstruktur. Model *Software Development Life Cycle (SDLC)* air terjun (*Waterfall*) sering juga disebut model sekuensial linier (*Sequential Linier*). Model air terjun (*Waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atauurut dimulai dari analisis, desain, pengkodean, dan pengujian (Rosa a.s., M.Shalahuddin, 2015). Tahapan *Waterfall* (Rosa a.s., M.Shalahuddin, 2015) yang dimaksud yaitu:

### 1. Analisis Kebutuhan Perangkat Lunak

Sebelum sistem dibuat, diperlukan suatu analisis sebagai dasar untuk mengetahui kebutuhan sistem ke depannya. Analisis kebutuhan sistem terdiri dari analisis kebutuhan fungsional yang bertujuan untuk mengetahui kebutuhan fungsi sistem dan analisis kebutuhan *non* fungsional untuk mengetahui perangkat keras dan perangkat lunak yang dibutuhkan serta kriteria pengguna sistem.

### 2. Desain

Desain berfungsi sebagai dasar perancangan yang mengubah data-data yang didapat dari analisis menjadi sebuah rancangan yang terdiri dari desain struktur data, struktur navigasi, dan rancangan antar muka.

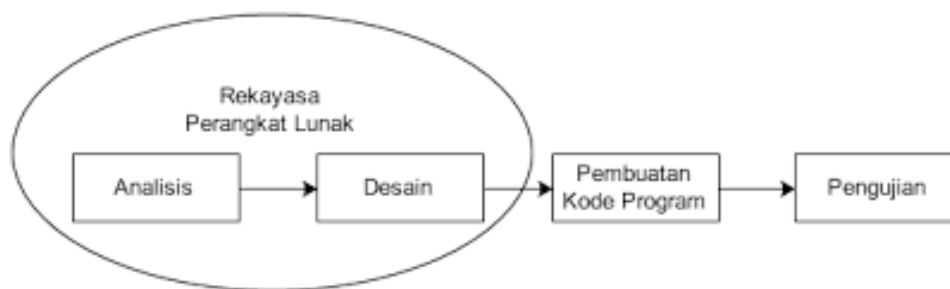
### 3. Pembuatan Kode Program

Tahapan ini merupakan lanjutan dari tahapan desain, yaitu implementasi desain menjadi sebuah program. Tahap ini menghasilkan suatu program yang sesuai dengan desain.

#### 4. Pengujian Program

Program yang telah dibuat wajib diuji terlebih dahulu untuk memastikan bahwa program layak digunakan dari segi *logic* maupun fungsional. Pengujian ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

Berdasarkan pendapat ahli di atas, dapat disimpulkan bahwa Model *Waterfall* merupakan Model SDLC paling sederhana yang cocok digunakan untuk pengembangan perangkat lunak dengan melakukan pendekatan secara sistematis dengan spesifikasi yang jarang berubah. Gambar 2.5 menunjukkan tahapan dari Model *Waterfall*.



Gambar 2. 5 Tahapan Metode *Waterfall*  
Sumber : (Rosa a.s., M.Shalahuddin, 2015, p. 28)

### 2.9 Metode *Blackbox Testing*

*Blackbox Testing* merupakan teknik pengujian perangkat lunak yang berfokus pada fungsionalitas dari perangkat lunak. *Blackbox Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Blackbox Testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program (Jaya, Januari 2018). Sederhananya, *Blackbox Testing*

atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Dengan menggunakan *Blackbox Testing* pengujian dilakukan berdasarkan sudut pandang pengguna. Hal tersebut dilakukan agar dapat menemukan inkonsistensi dalam perangkat lunak (Setiawan, Black Box Testing Untuk Menguji Perangkat Lunak, 2021).

## 2.10 Metode *System Usability Scale* (SUS)

*System Usability Scale* (SUS) merupakan metode evaluasi kegunaan yang memberikan hasil yang memadai berdasarkan pertimbangan jumlah sampel yang kecil, waktu, dan biaya. Hasil dari perhitungan dengan menggunakan metode ini akan dikonversi ke dalam sebuah nilai, yang dapat dijadikan pertimbangan untuk menentukan apakah sebuah aplikasi layak atau tidak layak untuk diterapkan (Ajie Wibowo Soejono, Arief Setyanto, Amir Fatah Sofyan, Maret 2018). Gambar 2.6 menunjukkan instrumen pengujian *System Usability Scale* (SUS).

| No | Pernyataan                                                                        | Skala |
|----|-----------------------------------------------------------------------------------|-------|
| 1  | Saya akan ingin lebih sering menggunakan aplikasi ini                             | 1 - 5 |
| 2  | Saya merasa aplikasi ini tidak harus dibuat serumit ini                           | 1 - 5 |
| 3  | Saya pikir aplikasi mudah untuk digunakan                                         | 1 - 5 |
| 4  | Saya membutuhkan bantuan dari orang teknis untuk menggunakan aplikasi ini         | 1 - 5 |
| 5  | Saya menemukan fitur pada aplikasi terintegrasi dengan baik                       | 1 - 5 |
| 6  | Saya pikir ada ketidaksesuaian dalam aplikasi ini                                 | 1 - 5 |
| 7  | Saya merasa kebanyakan orang mudah untuk mempelajari aplikasi dengan sangat cepat | 1 - 5 |
| 8  | Saya menemukan, aplikasi sangat rumit untuk digunakan                             | 1 - 5 |
| 9  | Saya percaya diri untuk menggunakan aplikasi ini                                  | 1 - 5 |
| 10 | Saya perlu belajar sebelum saya menggunakan aplikasi                              | 1 - 5 |

Gambar 2. 6 Instrumen Pengujian Metode SUS

Sumber: (Usman Ependi, Tri Basuki Kurniawan, Febriyanti Panjaitan, April 2019)

Seperti yang terlihat pada Gambar 2.6 dalam instrumen pengujian *System Usability Scale* (SUS) terdapat skala penilaian yang menjadi ukuran pembobotan ketika dilakukan pengujian. Skala penilaian tersebut diawali dengan 1 sampai dengan 5, 1 menunjukkan bahwa pengguna sangat tidak setuju dengan pernyataan pengujian dan 5 sangat setuju terhadap pernyataan pengujian (Usman Ependi, Tri Basuki Kurniawan, Febriyanti Panjaitan, April 2019). Gambar 2.7 menunjukkan skala penilaian dari pilihan jawaban *System Usability Scale* (SUS).

| Jawaban                   | Skor |
|---------------------------|------|
| Sangat Tidak Setuju (STS) | 1    |
| Tidak Setuju (TS)         | 2    |
| Ragu-Ragu (RG)            | 3    |
| Setuju (S)                | 4    |
| Sangat Setuju (SS)        | 5    |

Gambar 2. 7 Skala Penilaian Metode SUS

Sumber: (Saputra, November 2019)

Setelah melakukan pengumpulan data dari responden, data tersebut akan dihitung. Berikut aturan-aturan konversi tanggapan responden (Saputra, November 2019):

a). Pernyataan ganjil, yaitu 1, 3, 5, 7, dan 9 skor yang diberikan oleh responden dikurangi dengan 1.

$$\text{Skor SUS ganjil} : \sum Px - 1$$

Dimana  $Px$  adalah jumlah pertanyaan ganjil.

b). Pernyataan genap, yaitu 2, 4, 6, 8, dan 10 skor yang diberikan oleh responden digunakan untuk mengurangi 5.

$$\text{Skor SUS genap} : \sum 5 - Pn$$

Dimana  $Pn$  adalah jumlah pertanyaan genap.



c). Hasil dari konversi tersebut selanjutnya dijumlahkan untuk setiap responden kemudian dikalikan dengan 2,5 agar mendapatkan rentang nilai antara 0-100.

$$(\sum skor\ ganjil + \sum skor\ genap) \times 2,5$$

d). Setelah skor dari masing-masing responden diketahui, langkah selanjutnya adalah mencari skor rata-rata dengan cara menjumlahkan semua hasil skor dan dibagi dengan jumlah responden yang ada. Perhitungan ini dapat dilihat dengan rumus sebagai berikut:

$$\bar{X} = \frac{\sum x}{n}$$

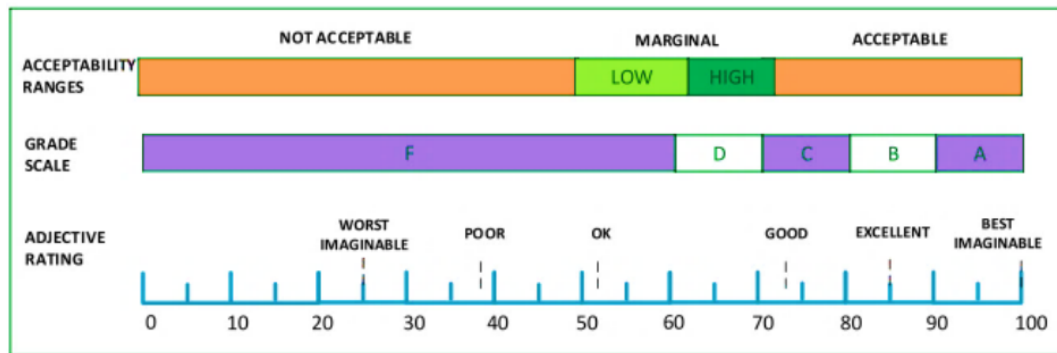
Keterangan:

$\bar{X}$  = Skor rata-rata

$\sum x$  = Jumlah skor *System Usability Scale*

n = Jumlah responden

Dari hasil tersebut akan diperoleh suatu nilai rata-rata dari seluruh penilaian skor responden. Untuk menentukan *grade* hasil penilaian ada 2 cara yang dapat digunakan. Penentuan pertama dilihat dari sisi tingkat penerimaan pengguna, yaitu *Grade Scale* dan *Adjective Rating* yang terdiri dari tingkat penerimaan pengguna terdapat 3 kategori yaitu *Not Acceptable*, *Marginal* dan *Acceptable*. Sedangkan dari sisi tingkat *Grade Scale* terdapat 6 skala yaitu A, B, C, D, E dan F. Dan dari *Adjective Rating* terdiri dari *Worst Imaginable*, *Poor*, *Ok*, *Good*, *Excellent* dan *Best Imaginable*. Gambar 2.8 menunjukkan penentuan tingkat penerimaan pengguna.



Gambar 2. 8 Tingkat Penilaian Pengguna Metode SUS  
Sumber: (Saputra, November 2019)

Penentuan yang kedua dilihat dari sisi *Percentile Rank* (SUS skor) yang memiliki *Grade* penilaian yang terdiri dari A, B, C, D dan E. Penentuan hasil penilaian berdasarkan *System Usability Scale* (SUS) *Score Percentile Rank* dilakukan secara umum berdasarkan hasil perhitungan penilaian pengguna. Gambar 2.9 menunjukkan penentuan *Percentile Rank*.

| Grade | Keterangan                  |
|-------|-----------------------------|
| A     | skor $\geq 80,3$            |
| B     | skor $\geq 74$ dan $< 80,3$ |
| C     | skor $\geq 68$ dan $< 74$   |
| D     | skor $\geq 51$ dan $< 68$   |
| E     | skor lebih $< 51$           |

Gambar 2. 9 *Percentile Rank Score* Metode SUS  
Sumber: (Saputra, November 2019)