

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Definisi aplikasi memiliki tujuan utama dari aplikasi untuk mengatur dan menjadwalkan waktu penggunaan *smartphone*. Aplikasi yang memiliki 3 fitur. Menu “atur waktu” yang digunakan untuk mengatur waktu dalam penggunaan *smartphone*, pilihan waktu penggunaan sesuai dengan usia anak menurut WHO yaitu usia 3-5 tahun dengan waktu 1jam/hari, dan 6-10 tahun dengan 3 pilihan waktu[10].

Pengaruh penggunaan *Smartphone* terhadap aktifitas kehidupan siswa menyimpulkan dampak negatif penggunaan *smartphone* terhadap kehidupan siswa, disarankan untuk mencari solusi yang dapat mengurangi dampak negatif tersebut. Dari permasalahan diatas solusinya dengan cara memberikan *workshop* tentang *internet parental control* dan pengenalan aplikasi *parental control* pada *smartphone* yang dijadikan aplikasi pembantu dalam proses pengawasan penggunaan *smartphone* oleh anak dan juga pembatasan pada aplikasi ataupun *website* yang dapat diakses dalam waktu penggunaan *smartphone*[11].

2.2 Landasan Teori

2.2.1 UML (Unified Modelling Language)




UML (*Unified Modelling Language*) adalah bahasa


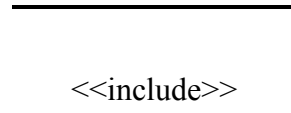
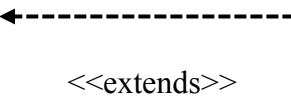
pemodelan visual yang digunakan oleh para pengembang untuk membuat blueprint atau rancangan sistem yang mudah dimengerti serta dilengkapi dengan mekanisme efektif untuk *sharing* dan mengkomunikasikan rancangan tersebut dengan yang lain. UML memudahkan dalam merancang *software* berorientasi objek[12].

a. *Use Case Diagram*

Use case Diagram merupakan pemodelan untuk melakukan (*beviar*) sistem informasi yang dibuat. *Use Case Diagram* menggambarkan interaksi antara *use case* (sistem) dan *actor*. Pada aplikasi *learning* menggunakan *moodle*[12].

Tabel 2.1 Simbol *Use Case*




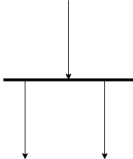
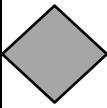

No	Simbol	Keterangan
1		<i>Use Case</i> : Abstraksi dan interaksi antarsistem dan <i>actor</i> .
2		<i>Actor</i> : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
3		Asosiasi antara <i>actor</i> dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi

		secara langsung dan bukannya mengindikasikan data.
4		Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
5		<i>Include</i> , Menunjukkan bahwa satu use case mengikutsertakan atau mencakup fungsionalitas dari use case lain (use case inkludir)
6		<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi tertentu terpenuhi.

b. Activity Diagram

Activity Diagram merupakan salah satu cara untuk memodelkan event yang terjadi dalam suatu *use case*. *Activity diagram* juga melakukan pengkodean proses pada sistem [12].

Tabel 2.2 *Activity Diagram*

No	Simbol	Keterangan
1		<i>Start Point</i> , titik awal dari suatu aktivitas atau proses dalam diagram
2		<i>End Point</i> , titik akhir dari suatu aktivitas atau proses dalam diagram alur kerja.
3		<i>Activities</i> , menggambar Tindakan atau langkah-langkah yang dilakukan dalam suatu proses atau kegiatan bisnis.
4		<i>Fork</i> /percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5		<i>Decision Points</i> , menggambarkan alur kerja dimana pengambilan keputusan terjadi (biasanya “ <i>true</i> ” atau “ <i>false</i> ”).
6		<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa yang bertanggung jawab atas setiap aktivitas.

2.2.2 Mobile

Aplikasi *mobile* juga dikenal sebagai *web app*, *online app*, *iPhone app* atau *smartphone app*. Aplikasi navigasi berbasis *mobile* merupakan solusi yang dapat menjawab permasalahan yang dijabarkan sebelumnya. Salah satu contoh perangkat *mobile* yang bisa digunakan adalah *smartphone* bersistem operasi *Android*. Ada tiga buah teknologi yang dapat dipadukan untuk membangun aplikasi navigasi yang dapat membantu teknisi untuk mencari alamat pelanggan[13].

2.2.3 Aplikasi Mobile

Aplikasi *mobile* adalah program yang disiapkan untuk melakukan fungsi tertentu dan dapat diinstal pada perangkat bergerak, yang secara luas merujuk pada kemampuan untuk berpindah dari lokasi satu ke lokasi lainnya. Secara lebih rinci, aplikasi *mobile* adalah perangkat lunak yang telah dipersiapkan untuk menjalankan fungsi tertentu dan dapat diinstal pada perangkat bergerak seperti *smartphone*[13]. Aplikasi *mobile* merupakan aplikasi yang dirancang khusus untuk platform *mobile* (misalnya *IOS*, *android*, atau *windows mobile*). Aplikasi *mobile* yang membantu para pengguna untuk dapat terkoneksi dengan layanan *internet* yang biasa diakses pada *PC*. *Mobile* merupakan kata sifat yang berarti dapat bergerak atau dapat digerakkan dengan bebas dan mudah[14].

2.2.4 Flutter SDK



Gambar 2.1 Logo Flutter

Flutter merupakan SDK untuk perkembangan aplikasi *mobile* dengan kinerja tinggi, aplikasi untuk *IOS* dan *android*, dari satu *codebase* (basis kode) yang dibuat oleh Google dengan lisensi *open source*. Sebelum resmi diluncurkan pada Desember 2018, Flutter dikembangkan oleh *Google* sejak 2015. Lalu, di tahun 2019, popularitas Flutter mulai meroket dan banyak *developer* berbondong-bondong menggunakan Flutter untuk menghadirkan aplikasi yang berkinerja tinggi pada platform yang berbeda. Untuk membuat aplikasi Flutter, diperlukan bahasa *dart*. Flutter mempunyai dua macam *widged* untuk perkembangan aplikasi pakai, yaitu *Material Design* dan *Cupertino*. Pada OS perangkat berbeda, *widged* bisa dipakai secara *cross platform*[15].

2.2.5 Visual Studio Code



Gambar 2.2 Logo Visual Studio Code

Visual Studio Code adalah aplikasi teks editor yang

dikembangkan oleh *Microsoft*. Berbagai sistem operasi seperti *Linuk*, *Mac*, dan *windows*. *Visual Studio Code* mendukung langsung beberapa Bahasa pemrograman populer, termasuk *JavaScrip*, *TypeScrip*, *Node Js*, dan Bahasa pemrograman lainnya. *Visual Studio Code* memudahkan para programmer saat berganti Bahasa pemrograman tanpa perlu mengganti aplikasi teks editor saat memahami dan konfigurasi *tools* kembali di aplikasi editor barunya. *Visual Studio Code* juga memberikan kebebasan kepada pengguna dalam tema, *debugger*, *extension*, dan lainnya[16].

2.2.6 Adobe XD



Gambar 2.3 Logo Adobe XD

Adobe XD UCD (*User Centered Design*) program yang diciptakan untuk memudahkan desainer dalam mengembangkan UI/UX aplikasi *mobile* atau *web*. *Software* ini mendukung *wireframing* situs *web* dan pembuatan prototipenya. Adobe menyediakan penawaran gratis pada Adobe XD untuk membuat UI/UX, dan *prototyping* ke dalam berbagai platform meliputi *web*, ponsel, tablet, dan sejenisnya[17].

2.2.7 Waterfall Model

Model *waterfall* pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi

merupakan model paling banyak dipakai dalam *software Engineering* (se) Metode air terjun yang sering disebut metode *waterfall* atau siklus hidup klasik (*Classic life cycle*), Model ini sebenarnya dikenal sebagai “*Waterfall Model*” atau “Model Air Terjun”. Model ini menggambarkan pendekatan yang sistematis dan berurutan dalam pengembangan perangkat lunak, dimulai dari spesifikasi kebutuhan pengguna dan berlanjut melalui tahap perancangan (*planning*), permodelan (*modelling*), konstruksi (*construction*), hingga penyerahan sistem kepada pengguna (*deployment*), dengan dukungan pada perangkat lunak[18]

2.2.8 Pengujian Sistem

1. *White Box Testing*

Pengujian *white box* digunakan berdasarkan sumber kode sehingga dapat mengidentifikasi bagian kode yang tidak berfungsi dengan benar. Beberapa keuntungan dari *white box testing* termasuk kemampuan untuk menghilangkan bagian yang tidak diperlukan dari kode yang tersembunyi, melakukan pengujian karena mengeksplorasi seluruh struktur atau logika, membantu dalam mengoptimalkan kode, dan memungkinkan pengujian dimulai meskipun antarmuka pengguna (GUI) masih dalam tahap pengembangan[19]. Dalam melakukan pengujian pada aplikasi Smart Timer dengan metode *white box testing* pengujian akan melihat kode secara keseluruhan kemudian

melakukan *testing* terhadap setiap kode yang ada jika kode tidak berfungsi maka bisa dihapus atau digantikan dengan kode yang difungsikan sebagaimana mestinya.

2. *Black Box Testing*

Pengujian black box, juga dikenal sebagai pengujian perilaku, melibatkan penguji yang tidak mengetahui struktur internal atau logika perangkat lunak yang diuji. Pengujian hanya bergantung pada spesifikasi kebutuhan dan tidak perlu menganalisis kode. Keuntungan dari pengujian black box termasuk kemampuannya untuk menemukan aspek yang tidak sesuai dengan spesifikasi kebutuhan yang diberikan dalam pengembangan perangkat lunak, kelemahan dari pengujian black box adalah penguji tidak dapat dilakukan secara menyeluruh karena keterbatasan pengetahuan penguji tentang perangkat lunak[19]. Dalam pengujian pada aplikasi *Smart Timer* menggunakan metode *black box testing* penguji hanya melihat *interface* aplikasi kemudian dilakukan pengujian pada fungsi dari spesifikasi aplikasi.