

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penelitian perancangan aplikasi menggunakan metode *waterfall* sudah banyak dibahas serta digunakan. Seperti penelitian berjudul “Penerapan Metode *Waterfall* dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android” yang dilakukan oleh Dini Silvi Purnia, Achmad Rifai, Syaifur Rahmatullah pada tahun 2019 yang bertujuan untuk membuat aplikasi android yang berfungsi sebagai pembantu sosial dengan metode *waterfall*, alasan penggunaan dalam penelitian ini dengan metode *waterfall* adalah dengan menggunakan metode ini tahapan dari penelitian ini dilakukan secara terurut karena metode yang dilakukan berurutan dan berkelanjutan, seperti sebuah air terjun. tahapan - tahapan model *waterfall*[11]. Hasil data angket setelah uji coba yang diberikan menghasilkan 40% sangat setuju, 30% setuju, 20% netral, 10 % tidak setuju, dan 0 % sangat tidak setuju.

Penelitian lain yang berjudul “*Application of the Waterfall Method on a Web-Based Job Training Management Information System at Trunojoyo University Madura*” yang dilakukan oleh Sri Herawati, Yudha Dwi Putra Negara, Husnul Fuadi Febriansyah, dan Doni Abdul Fatah. Pada jurnal ini dilakukan dengan pengujian, Pengujian dilakukan oleh sistem dengan 13 kali pengujian menggunakan black box dengan hasil yang valid[12].

Kemudian penelitian berjudul “Perancangan Aplikasi Game Edukasi Menggunakan Model *Waterfall*” oleh Suryadi Andri ada 5 tahap yang dilakukan saat memulai penelitian, Analisis kebutuhan perangkat lunak, desain, pengujian, dan pendukung (*support*) / pemeliharaan (*maintenance*). Metode black box juga digunakan di penelitian ini. Tahapan terakhir dalam model *waterfall* adalah tahapan

pengujian, dimana pada tahapan ini software yang telah dibuat diuji apakah sudah sesuai dengan kebutuhan atau belum[13].

Penelitian lain berjudul “Perancangan Aplikasi Pembelajaran Bahasa Jawa Berbasis Android” yang dilakukan oleh Ayu Ernawati, dan Aries Saifudin pembuatan aplikasi menggunakan Bahasa pemograman *JavaScript*, *HTML5*, *CSS*, *Ionic Framework*, dan *MySQL*. Dalam hal tersebut bagaimana aplikasi dapat digunakan secara *offline*, basis data yang diunduh dari webapp dan dapat diubah dari menjadi *sqlite*, kemudian disatukan ke dalam paket instalasi aplikasi android.[14] Pengujian aplikasi pun tetap menggunakan *black box* yang menunjukkan tidak ada kesalahan / cacat.

Selain itu ada pun penelitian berjudul “Rancang Bangun Aplikasi Dilib-Us (*Digital Library*) Berbasis Android Menggunakan Metode *Prototyping*” yang dilakukan oleh Yustina Meisella Kristania, dan Joko Dwi Mulyanto. Kali ini rancang bangun aplikasi yang akan dilakukan menggunakan metode *Prototype*. Langkah-langkah dalam penggunaan metode *Prototyping* adalah dengan analisis atau pengumpulan data kebutuhan, lalu melibatkan pengembang dan pengguna dalam sistem pengembangan untuk menentukan fungsi, kebutuhan operasional sistem dan tujuan[15].

Tabel 2. 1 Ringkasan Penelitian Terdahulu

No.	Judul	Isi	Tujuan	Kritik	Sinteisis	Ringkasan
1.	Penerapan Metode <i>Waterfall</i> dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android[11]	Perancangan Aplikasi untuk memudahkan bantuan sosial yang dapat terhubung langsung dengan badan hukum.	Membuat aplikasi yang agar para penyumbang dana serta badan hukum yang berurusan lebih mudah untuk pembagian sumbangan berupa barang ataupun uang	Aplikasi yang dikembangkan dibuat hanya untuk Android, serta tidak adanya <i>activity diagram</i>	Penelitian ini menggunakan metode <i>waterfall</i> sebagai metodologi penelitian dan menggunakan uji coba dengan metode <i>black box testing</i> .	Hasil dari perancangan aplikasi sistem informasi bantuan sosial ini terhubung dengan badan hukum secara langsung, pengujian <i>black box</i> dilakukan hasilnya valid semua.
2.	<i>Application of the Waterfall Method on a Web-Based Job Training</i>	Melakukan penelitian untuk merancang website untuk menejemen informasi tentang	Pembuatan aplikasi ini untuk memudahkan para mahasiswa di Kampus	Analisa dari perancangan website tidak menggunakan <i>activity diagram</i> . Pada	Pada penelitian ini penggunaan metodologi penelitian	Pada pengembangan website untuk pembantu kerja

No.	Judul	Isi	Tujuan	Kritik	Sinteisis	Ringkasan
	<i>Management Information System at Trunojoyo University Madura</i> [12]	kerja praktik yang akan dilakukan di Universitas Madura Trunojoyo	Universitas Madura Trunojoyo untuk infomrasi kerja praktik.	rancangan website juga tidak ada use interface untuk daftar	menggunakan metode <i>waterfall</i> , uji coba website tersebut menggunakan metode <i>Black Box Testing</i> .	praktik ini menguji 13 kali menggunakan metode <i>black box</i> dengan hasil yang valid dan sesuai
3.	Perancangan Aplikasi Game Edukasi Menggunakan Model <i>Waterfall</i> [13]	Penelitian ini melakukan perancangan aplikasi untuk pembuatan game edukasi dengan permainan ular tangga berisi pertanyaan seputar pengetahuan umum.	Pembuatan game ini bertujuan agar game ini diharapkan dapat membantu siswa memahami materi belajar dengan cara menyenangkan sehingga dapat meningkatkan pemahaman siswa.	Subjek untuk penelitian masih belum ada sehingga penerapan aplikasi game ini masih belumlah jelas	Metode yang digunakan disini adalah metode <i>waterfall</i> serta uji coba menggunakan <i>black box testing</i> .	Penggunaan Metode <i>waterfall</i> terhadap Perancangan aplikasi game ular tangga bertemakan pengetahuan umum lalu diuji dengan <i>black box</i> memiliki hasil yang valid.

No.	Judul	Isi	Tujuan	Kritik	Sinteisis	Ringkasan
4.	Perancangan Aplikasi Pembelajaran Bahasa Jawa Berbasis Android[14]	Penelitian ini dilakukan untuk merancang aplikasi pembelajaran Bahasa Jawa yang dapat digunakan oleh para siswa maupun guru.	Melakukan penelitian ini bertujuan agar para siswa untuk belajar tentang Bahasa Jawa	Metode penelitian yang digunakan masih kurang jelas hanya menggunakan Metode <i>black box</i> sebagai testingnya saja	Pengujian aplikasi ini menggunakan metode <i>black box</i>	Perancangan aplikasi pembelajaran Bahasa Jawa ini menggunakan <i>black box</i> sebagai metode pengujian, pengujian dilakukan 3 kali dan hasilnya valid
5.	Rancang Bangun Aplikasi Dilib-Us (<i>Digital Library</i>) Berbasis Android Menggunakan Metode <i>Prototyping</i> [15]	Perancangan Aplikasi untuk perpustakaan online yang berfungsi untuk meningkatkan mutu dari perpustakaan	Penelitian bertujuan agar dapat memudahkan pengguna serta perpustakaan dalam mencari meminjam, dan mengembalikan buku.	Tidak adanya metode pengujian yang dilakukan.	Menggunakan metode <i>prototype</i> serta uji coba dengan <i>black box testing</i> .	Pembuatan aplikasi perpustakaan online untuk memudahkan dalam pengelolaan perpustakaan menggunakan <i>prototype</i> sebagai

No.	Judul	Isi	Tujuan	Kritik	Sinteisis	Ringkasan
						metode yang hasil pengembannya diharapkan dapat dipakai dibanyak perpustakaan.

Berdasarkan pada table 2.1 Penelitian Terdahulu pengembangan aplikasi serta *website* menggunakan metode *waterfall* dan ada juga penerapan *metode prototyping* serta pengujian menggunakan metode *blackbox*. Di ambil dari lima penelitian terdahulu penggunaan metode *waterfall* digunakan di penelitian ini dan untuk pengujian pada aplikasi ini terlihat pada table 2.1 metode yang digunakan metode *blackbox*.

2.2 Dasar Teori

2.2.1 Kurikulum Holistik

Kurikulum Holistik memiliki konsep merupakan suatu konsep model dari kurikulum yang dirancang menggunakan prinsip keterpaduan atau dihubungkan dengan aspek-aspek (mata pelajaran dan lingkungan siswa), terbuka (akomodasi dan menjangkau berbagai kondisi siswa), dan keseimbangan (proses pembelajaran yang dirancang agar memenuhi seluruh potensi siswa, seperti intelektual, spiritual, dan emosional) secara seimbang[16].

Pada proses pembelajaran kurikulum holistik memiliki aset yang ditetapkan, menurut Nawono mengatakan bahwa dari 40 aset yang ada, sebagai pembentukan kepribadian anak harus ditentukan aset mana yang harus dicapai. Dalam proses pembelajaran ada ketentuan-ketentuan yang harus dipenuhi, seperti waktu yang diperlukan, perlengkapan dalam penyampaian materi, tempat yang sesuai dengan materi, dan penyampaian atau strategi dalam penyajian, serta yang terakhir adalah evaluasi pembelajaran[17].

Pelaksanaan pembelajaran Kurikulum Holistik dilakukan melalui 3 kegiatan, yang pertama tahapan pendahuluan, tahapan inti, serta tahapan penutup.

- A. Tahap pengenalan meliputi mempersiapkan siswa secara psikis dan fisik untuk mengikuti proses pembelajaran, memberikan pertanyaan dengan cara mengaitksn pengetahuan sebelumnya dengan materi yang akan dipelajari kemudian menjelaskan tujuan dari pembelajaran atau kompetensi dasar yang akan dicapai, dan menyampaikan ruang lingkup materi.
- B. Tahapan inti dilakukan dengan melihat karateristik siswa dan mata pelajaran, meliputi elaborasi, konfirmasi, ekspolrasi, dan pendekatan antara siswa dan guru.
- C. Ditahapan penutup guru bersama dengan siswa melakukan kegiatan untuk membuat kesimpulan pelajaran, dengan cara melakukan refleksi terhadap kegiatan yang sebelumnya suda dilaksanakan, memberikan umpan balik terhadap proses dan hasil.

2.2.2 Metode Waterfall

Metode *Waterfall* salah satu pendekatan dari *Software Development Life Cycle (SDLC)* paling pertama yang dapat digunakan untuk pengembangan perangkat lunak[18]. Metode *Waterfall* memiliki proses yang berurutan yang dimulai dari proses analisis kebutuhan, desain sistem, implementasi pengkodean, lalu pengujian. Tahapan-tahapan dari metode *waterfall* harus dilaksanakan secara berurutan, disebut *waterfall* (Air Terjun) dikarenakan pengerjaan prosesnya harus dilalui langkah demi langkah dan tidak dapat diloncati untuk ketahap berikutnya. Berikut penjelasan dari tahapan dari metode *waterfall*.

1. Analisis Kebutuhan

Dalam tahap awal yaitu, analisis kebutuhan dilakukan dengan cara pengumpulan data analisa dari user seperti apa aplikasi yang diinginkan untuk di spesifikasikan agar memahami aplikasi seperti apa yang akan dirancang.

2. Desain Sistem

Lalu pada tahap kedua yaitu, desain sistem dilakukan setelah tahap analisis kebutuhan selesai, proses desain akan berfokus pada rancangan antar muka, aritektur perangkat lunak, serta prosedur saat akan melakukan pengkodean. Ditahap ini mentranslasi dari analisis kebutuhan dibuat menjadi sebuah desain agar dapat diimplementasikan menjadi sebuah program.

3. Pengkodean

Setelah tahap pembuatan desain dilakukan, desain dibuat ke dalam program perangkat lunak dengan pengkodean yang sesuai pada desain yang sudah dibuat.

4. Pengujian

Pengujian difokuskan kepada program aplikasi yang sudah dibuat secara fungsional dan menguji semua bagian agar meminimalisir error serta kesesuaian dengan program yang dijalankan. Cara pemilihan pengujian

dapat dilakukan menggunakan data yang akan sering digunakan dimulai dari input dan output, pengolahan data, data operasional.

2.2.3 Android

Android merupakan salah satu platform penyedia *platform* yang terbuka bagi pengembang agar dapat menciptakan suatu aplikasi yang digunakan oleh bermacam perangkat lunak. Saat ini Android adalah sebuah sistem operasi mobile yang terpopuler di dunia[19].

Android adalah sistem operasi yang dirancang Google yang berbasis *kernel Linux* dan berbagai perangkat lunak *Open Source* lainnya dan biasa digunakan untuk perangkat dengan layar sentuh seperti *smartphone* dan tablet[20].

Android pertama kali dirilis secara resmi pada September 2008 sebagai android versi 1.0. Untuk versi terbaru Android adalah Android versi 12 dirilis dianggal 12 Oktober 2021.

2.2.4 iOS

iOS merupakan *iPhone Operating system*, yang memiliki sistem operasi dari Unix digunakan dalam semua perangkat *Apple* seluler. Awalnya iOS memiliki nama lain, yaitu iPhone OS. Untuk versi pertama dari iOS diumumkan bersama dengan rilis iPhone pertama pada 9 Januari 2007[21].

Keunggulan dari iOS adalah memiliki tingkat keamanan lebih baik dari android serta control yang lebih ketat. Dikarenakan iOS adalah *proprietary software*, sehingga hanya tim pengembang Apple yang memiliki dan mampu memodifikasi source code iOS yang membuat performa iOS menjadi sangat baik dan sangat aman.

2.2.5 *Flutter*

Flutter pertama kali di luncurkan pada tahun 2015 pada *Dart Developer Summit* dengan nama Sky[22]. *Flutter* adalah kerangka kerja *UI portabel* Google untuk membangun aplikasi *modern, native*, dan reaktif untuk iOS dan Android. Google juga bekerja pada *Flutter desktop, Flutter web* dan juga *embedded device* (Raspberry Pi, home, otomotif, dan banyak lagi).

Flutter sebagai *framework* menggunakan Dart sebagai bahasa pemrograman berorientasi objek modern yang dikompilasi ke pada ARM dan *JavaScript*. *Flutter* merendering dengan Skia 2D yang disponsori oleh Google dan berbagai platform lain seperti perangkat keras dan perangkat lunak dan juga dapat digunakan oleh, Android, iOS, dan lainnya[23].

2.2.6 *Application Programming Interface (API)*

Application Programming Interface atau API merupakan penghubung yang menghubungkan antara satu pihak dengan pihak lain agar bisa saling terhubung. Pihak – pihak ini bisa berbentuk perangkat lunak ataupun perangkat keras. API berisi prosedur, fungsi, cara komunikasi atau peralatan untuk komunikasi.

API merupakan integrasi dari dua bagian dari sistem aplikasi. Terdiri dari elemen *function, protocols, dan tool* lainnya digunakan untuk pengembangan membuat sebuah aplikasi[24].

2.2.7 *Dart*

Dart adalah bahasa pemrograman yang saat ini dikembangkan oleh Google dan diresmikan pada 10 Oktober 2011. *Dart* dikembangkan untuk pembuatan aplikasi dengan berbagai platform dengan lebih optimal. *Dart* sebagai Bahasa pemrograman biasa digunakan dalam pembuatan aplikasi atau dalam bentuk, *web, command line interface* ataupun *mobile (Android dan iOS)*[25].

2.2.8 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah suatu bahasa standar biasa digunakan dalam pendokumentasian sistem, dapat dikenal sebagai bahasa standar dalam penulisan untuk membuat pemodelan secara visual dapat digunakan untuk sarana perancangan dalam sistem berorientasi objek[26]. Diagram UML 2.5 jenis klasifikasi dari diagram UML dibagi dua kategorai, diagram struktur dan diagram perilaku.

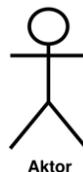
Pada penelitian ini, akan menggunakan diagram UML dengan kategori perilaku (behavior diagrams) yaitu use case diagram, activity, dan sequence diagram :

1. *Usecase Diagram*

Usecase diagram untuk menggambarkan serangkaian tindakan yang harus atau dapat dilakukan oleh beberapa sistem (subjek) dalam berkolaborasi dengan satu atau lebih pengguna eksternal sistem (aktor)[27]. Berikut adalah beberapa elemen dari diagram use case:

a. Aktor

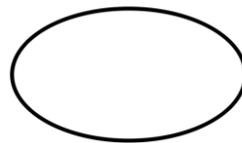
Aktor menggambarkan pengguna, sistem lain yang berinteraksi langsung dengan sistem atau fungsi.



Gambar 2. 1 Notasi Aktor[27]

b. *Use case*

Use case merupakan elemen gambaran dari fungsional dalam sistem, sehingga pembuat maupun konsumen dapat mengerti mengenai alur dari sistem yang akan dibuat. *Use case* dinotasikan dalam Gambar 2.2.



Gambar 2. 2 Notasi Use case[27]

c. Hubungan asosiasi

Asosiasi merupakan gambaran abstraksi untuk penghubung antara aktor dengan use case. Hubungan asosiasi dinotasikan dalam Gambar 2.3.

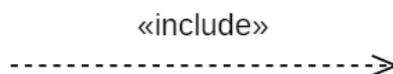
s



Gambar 2. 3 Notasi hubungan Asosiasi[27]

d. Hubungan *include*

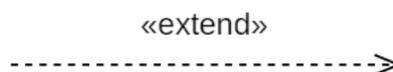
Hubungan *include* menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya. Hubungan *include* dinotasikan dalam Gambar 2.4.



Gambar 2. 4 Notasi hubungan *include*[27]

e. Hubungan *exclude*

Hubungan *exclude* memperlihatkan untuk suatu *use case* yang merupakan tambahan dari fungsional *use case* lainnya jika kondisi tertentu terpenuhi.



Gambar 2.18 Notasi hubungan *exclude*[27]

2. *Activity Diagram*

Activity diagram berfungsi untuk menunjukkan urutan dan kondisi untuk mengoordinasikan perilaku tingkat yang lebih rendah, daripada pengklasifikasi mana yang memiliki perilaku tersebut. Ini biasanya disebut *control flow* (aliran kontrol) dan *object flow models* (model aliran objek) [27]. Berikut adalah beberapa elemen dari *activity diagram*:

a. *Initial node*

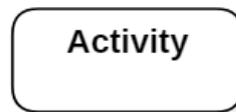
Initial node merupakan elemen untuk menandakan sebuah status awal dari aktivitas sebuah sistem.



Gambar 2. 5 Notasi *Initial Node*[27]

b. *Activity*

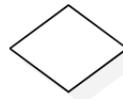
Activity adalah suatu elemen yang dapat menandakan aktivitas pada suatu sistem. *Activity* dinotasikan seperti pada Gambar 2.6.



Gambar 2. 6 Notasi *activity*[27]

c. *Decision*

Decision adalah suatu elemen yang biasa digunakan jika pilihan aktivitas ada lebih dari satu dengan kondisi tertentu. *Decision* dinotasikan dalam Gambar 2.7.



Gambar 2. 7 Notasi *decision*[27]

d. *Join*

Join adalah suatu elemen yang biasa digunakan dalam menggabungkan lebih dari satu aktivitas. *Join* dinotasikan pada Gambar 2.8.

Gambar 2. 8 Notasi *join*[27]

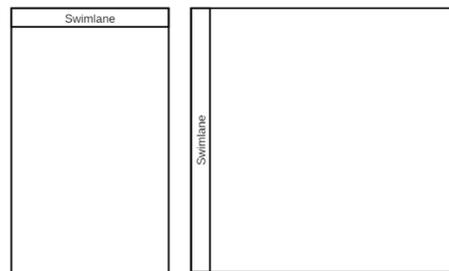
e. *Final node*

Final node adalah elemen yang menunjukkan status akhir pada aktivitas sebuah system. *Final node* dinotasikan dalam gambar 2.9.

Gambar 2. 9 Notasi *final node*[27]

f. *Swimlane*

Swimlane adalah wadah pada setiap aktivitas. *Swimlane* dinotasikan pada Gambar 2.10.

Gambar 2. 10 Notasi *Swimlane*[27]

2.2.9 *Firestore*

Firestore merupakan salah satu platform manajemen basis data, data yang disimpan dalam *Firestore* bersifat *realtime*. *Firestore* menyediakan rangkaian pustaka yang komprehensif untuk sebagian besar *platform web dan mobile*. Selain itu, *Firestore* juga dapat diintegrasikan dengan berbagai *framework* lain seperti *node, java*, dan sejenisnya[28].

Firestore Realtime Database adalah sebuah database cloud yang menggunakan format JSON untuk menyimpan data dan mengatur sinkronisasi data secara real-time kepada setiap klien yang terhubung. Saat membangun aplikasi hybrid yang dapat berjalan di berbagai platform, seperti Android dan iOS, semua klien akan berbagi satu instance dari Realtime Database dan secara otomatis menerima pembaruan dengan data yang spesifik. Dibandingkan dengan basis data relasional, Firestore Realtime Database merupakan sebuah basis data NoSQL yang memiliki optimalisasi dan fungsionalitas yang berbeda[29].

2.2.10 Pengujian Aplikasi

Dalam pengembangan aplikasi atau perangkat lunak, pengujian adalah salah satu tahapan terpenting, terutama untuk memastikan bahwa kode perangkat lunak dapat diskalakan dan mudah dipelihara di masa mendatang. Selain itu, untuk memastikan semua fitur aplikasi berjalan dengan baik. Tujuannya adalah untuk memastikan bahwa aplikasi atau perangkat lunak berjalan dengan baik dan tidak mengalami kecacatan atau malfungsi yang dapat memengaruhi keamanan dan kenyamanan pengguna. Dalam penelitian ini, metode pengujian yang digunakan adalah *Black Box*.

2.2.10.1 Metode *Black Box*

Black box testing, juga dikenal sebagai *Behavioral Testing*, merujuk pada metode pengujian di mana observasi dilakukan terhadap input dan output perangkat lunak tanpa memiliki pengetahuan tentang struktur kode yang ada di dalamnya [30]. Ada beberapa teknik yang umum digunakan dalam pengujian *Black Box*, seperti *All pair testing*, *Boundary value analysis*, *cause-effect graph*, *Equivalence partitioning*, *fuzzing*, *Orthogonal array setting*, dan *State transition*.

Penggunaan *Black Box* memiliki lebih banyak keuntungan daripada kerugiannya. Beberapa keuntungan menggunakan *Black Box* termasuk tidak perlu mengetahui kode program untuk melakukan pengujian, pengujian perangkat lunak dilakukan dengan mengamati hasil

eksekusi melalui data uji, dan memeriksa fungsionalitas perangkat lunak.[31]

Pada persamaan 2.1 dihitung presentasi kelayakan dalam hitungan persen setelah uji skenario dengan menggunakan pengujian *blackbox*.

$$\text{Presentasi Kelayakan (\%)} = \frac{\text{total valid test scenario}}{\text{total test scenario}} * 100 \quad (2.1)$$