

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Untuk meningkatkan perolehan informasi yang terkait penelitian ini, diperlukan studi literatur dari penelitian terkait yang menggunakan framework Flutter dan pengembangan aplikasi pembelajaran huruf dan Bahasa Korea yang dapat digunakan sebagai bahan untuk melengkapi data. Beberapa jurnal dan artikel berikut dipilih berdasarkan tema yang relevan dengan penelitian

APLIKASI PEMBELAJARAN DUAL BAHASA KOREA DAN JEPANG BERBASIS ANDROID (2018)

Penelitian ini membahas tentang perancangan aplikasi untuk pembelajaran dual Bahasa yaitu Bahasa Korea dan Jepang yang meliputi pembelajaran bahasa dasar seperti pengenalan dan percakapan serta partikel penyusun kalimat. Pengembangan menggunakan web-view sehingga menggunakan HTML dan CSS dalam pembuatan antarmuka serta penggunaan SQLite untuk penyimpanan Data [11]

APLIKASI PEMBELAJARAN BAHASA KOREA BERBASIS ANDROID (2021)

Penelitian ini membahas tentang pembuatan aplikasi pembelajaran Bahasa Korea berbasis Android, penelitian ini berfokus pada pembelajaran Huruf Korea (Hangeul) dan dengan tujuan untuk membantu pengguna yang memiliki waktu serta biaya yang terbatas sehingga bisa membantu pembelajaran dalam memahami Bahasa Korea dengan biaya yang lebih murah, peneliti menggunakan SDLC (Software Development Life Cycle), metode tersebut menyediakan pendekatan alur hidup perangkat lunak secara terurut. [12]

APLIKASI PEMBELAJARAN BAHASA KOREA MENGENAL HURUF DAN ANGKA KOREA (HANGEUL) UNTUK USIA 18-20 TAHUN BERBASIS ANDROID (2017)

Penelitian ini membahas mengenai pengembangan aplikasi pembelajaran Aksara Korea Hangeul dan angka untuk remaja umur 18-20 tahun menggunakan system operasi Android. Metode pengerjaan menggunakan model ADDIE(Anaysis, Design, Develop, Implement, Evaluate) dengan berfokus pada pembelajaran berbasis audio atau *tone* pengucapan huruf *hangeul*. [13]

IMPLEMENTASI DESAIN PATTERN BLOC DAN PATTERN REPOSITORY PADA APLIKASI ENERGY MANAGEMENT SYSTEM PT.PLN BERBASIS MOBILE DENGAN FLUTTER (2020)

Penelitian ini membahas tentang pengimplementasian Teknik desain untuk membangun sebuah aplikasi berbasis mobile dengan flutter berdasarkan studi kasus di PT.Indonesia Comnets Plus. Aplikasi ini diharapkan dapat mendukung kegiatan operasional dan memudahkan monitoring pemaikaian listrik untuk PLN group.Metode yang diterapkan adalah metode desain pattern berupa pattern BLoC [9]

PERANCANGAN APLIKASI MOBILE LEARNING HURUF HANGEUL KOREA (2020)

Penelitian ini mengenai perancangan aplikasi *mobile learning* huruf *hangeul* Korea untuk sitem operasi android, aplikasi ini berfokus pada pengenalan huruf Korea, cara membaca serta cara penulisan huruf *hangeul* yang benar. Pembelajaran melalui aplikasi *mobile* dianggap lebih interaktif dan menyenangkan dan tidak harus mengeluarkan banyak biaya [14]

| No | 1 | 2 | 3 | 4 | 5 |
|-------------------|---|--|---|--|---|
| Judul | Aplikasi Pembelajaran Dual Bahasa Korea Dan Jepang Berbasis Android | Aplikasi Pembelajaran Bahasa Korea Berbasis Android | Aplikasi Pembelajaran Bahasa Korea Mengenal Huruf Dan Angka Korea (Hangeul) Untuk Usia 18-25 Tahun Berbasis Android | Implementasi Desain Pattern Bloc Dan Pattern Repository Pada Aplikasi Energy Management System Pt.Pln Berbasis Mobile Dengan Flutter | Perancangan Aplikasi Mobile Learning Huruf Hangeul Korea |
| Tahun | 2018 | 2021 | 2017 | 2020 | 2020 |
| Kontribusi | Melakukan pengembangan aplikasi pembelajaran dual bahasa Korea dan Jepang berbasis menggunakan <i>Basic4Android</i> | Merancang aplikasi pembelajaran bahasa Korea berbasis Android dengan metode SDLC | Pengembangan Aplikasi pembelajaran bahasa Korea pada sistem operasi Android | Melakukan pengembangan aplikasi <i>energy management system</i> untuk PT. PLN dengan menggunakan Flutter | Melakukan analisis dan perancangan aplikasi pembelajaran huruf <i>hangeul</i> berbasis <i>android</i> |
| Metode | Studi Pustaka dan wawancara | SDLC | ADDIE (<i>Analysis, Design, Develop, Implement, Evaluate</i>) | Desain <i>Pattern BloC</i> dan <i>Repository pattern</i> | UAT (<i>User Acceptance Test</i>) |
| Limitasi | Penelitian ini hanya menjelaskan proses analisa kebutuhan sistem dan tidak menjelaskan bagaimana proses | Pada penelitian ini tidak dibahas mengenai pengujian sistem. | Pengembangan fitur aplikasi dapat dikatakan sederhana, hanya memiliki fitur | Sistem monitoring pada aplikasi belum dikembangkan secara sempurna, terkendala | Pengembangan fitur aplikasi dapat dikatakan sederhana, hanya memiliki fitur dasar |

| | | | | | |
|--------------|---|--|---|---|--|
| | pengembangan aplikasi dilakukan | | dasar sebuah aplikasi pembelajaran bahasa | perizinan dari pihak PT. PLN selaku <i>stakeholder</i> | sebuah aplikasi pembelajaran bahasa |
| Hasil | Aplikasi ini membahas tentang pembelajaran dual bahasa sekaligus dimana pengguna mampu memahami konsep dasar bahasa Korea dan Jepang yang memiliki persamaan dalam struktur kalimat dan penggunaan tata bahasa. Pembelajaran huruf, percakapan sehari-hari baik formal maupun non formal, kosakata pembantu untuk dihafalkan serta soal latihan yang diberikan diharapkan mampu membantu dalam proses pemahan dua bahasa sekaligus. | Aplikasi ini memberikan kemudahan bagi pengguna karena aplikasi ini dapat digunakan tanpa internet. Rancangan aplikasi yang dibuat dan juga dari segi penggunaan aplikasi dapat dengan mudah dipahami. | Aplikasi Pembelajaran ini berfokus pada pengenalan huruf dan angka Korea dengan pemilihan kategori umur pengguna pada rentang 18-25 tahun | Dengan menggunakan <i>Pattern</i> BloC dan juga <i>Repository</i> menghasilkan aplikasi dengan performa yang lebih efisien dan mengurangi penggunaan memori secara signifikan sehingga dapat mengurangi kebocoran memori. | Menghasilkan aplikasi yang menarik dan memiliki aspek yang cukup lengkap sehingga memudahkan pengguna untuk menggunakannya sebagai media pembelajaran huruf <i>hangeul</i> |

2.2 Dasar Teori

2.2.1 Flutter

Flutter adalah SDK atau framework yang dikembangkan oleh Google pada 2015 dan kemudian terus dikembangkan dan mendapatkan response yang baik dan akhirnya rilis stabil pada 2018. Tujuan utamanya adalah untuk membuat sesuatu yang lebih baik pada aplikasi *mobile* utamanya pada saat proses pengembangan dan *user experience* serta *rendering* dengan performa yang tinggi, membuat aplikasi native untuk Android dan iOS dengan satu basis kode saja.[22]

Kelebihan Flutter dibanding *framework cross-platform* lainnya yaitu memiliki performa yang tinggi, kodenya akan langsung diproses langsung oleh Dart AOT yang akan berjalan pada Dart virtual machine dan kemudian diubah langsung ke kode native , serta tidak menggunakan widget Original Equipment Manufacturer(OEM) untuk tampilannya, melainkan menggunakan mesin rendernya sendiri yaitu Skia yang sudah didukung *OpenGL ES2-Accelerated backend*, dengan begitu *developer* bisa menciptakan widget sendiri yang unik dan tidak dibatasi aturan seperti ketika menggunakan OEM.[22] Selain itu, flutter dikembangkan secara ringkas dan cepat dengan *hot reload*, yang membantu perangkat asli emulator untuk langsung menampilkan perubahan kode. Dalam beberapa detik, kode yang dimodifikasi akan dimuat ulang saat aplikasi berjalan tanpa memulai ulang. Ini bagus tidak hanya untuk membuat antarmuka pengguna dan menambahkan fitur, tetapi juga untuk *debugging* [24]. Flutter adalah *open source* yang sangat baik, memiliki dokumentasi komprehensif yang sangat baik, dan memiliki komunitas untuk membantu memecahkan masalah yang ditemukan selama pengembangan [22]

2.2.2 BLoC

BLoC atau *business Logic Component* adalah *design pattern* dan juga *state management* untuk Flutter. Memiliki Konsep hampir sama dengan MVVM (Model - View - ViewModel), yaitu menjadi penengah antara tampilan antarmuka(UI) dan juga layer data, atau yang sering disebut juga repository,

dengan begitu akan membagi sebuah struktur aplikasi menjadi tampilan antarmuka(UI), BloC dan bagian pemrosesan data. Prinsipnya adalah BloC menerima event dalam hal ini bisa berupa perubahan state/keadaan atau interaksi pengguna misal menekan suatu tombol dan kemudian akan memprosesnya sesuai dengan business logic yang telah ditulis kodenya yang kemudian akan mengembalikan sebuah hasil berupa state/keadaan yang akan terus dipantau oleh widget tampilan(UI) sehingga akan bereaksi ketika mendeteksi adanya perubahan state [25],

2.2.3 Desain Pattern

Dalam pembuatan kode program, *programmer* terkadang menggunakan suatu metode desain yang hanya dapat menyelesaikan satu permasalahan tetapi ketika metode desain itu digunakan pada permasalahan yang lebih rumit, metode tersebut menjadi susah untuk diimplementasikan sebagai teknik permasalahan. Solusi umum ini biasa disebut sebagai desain pattern[21]

2.2.4 Aksara Korea (Hangeul)

Hangeul adalah huruf atau aksara yang digunakan dalam menulis bahasa Korea. Diciptakan pada 1443 oleh raja keempat dimasa kerajaan Chosun yaitu Raja Agung Sejong, kemudian pada 1446 resmi dipublikasikan ke masyarakat Korea. Hangeul berjumlah 40 karakter yang terdiri dari 21 huruf vocal serta 19 huruf konsonan. Aksara *hangeul* terdiri dari 4 bagian yakni huruf vokal, huruf konsonan, huruf gabungan vokal serta huruf gabungan konsonan. [14]

Bentuk dari aksara hangeul sekilas mirip dengan aksara Cina, tetapi *hangeul* merupakan sebuah alfabet. Hal ini disebabkan karena setiap huruf adalah lambing huruf konsonan dan vokal yang berbeda.[14]

2.2.5 Framework

Framework adalah sebuah kerangka kerja pada pemrograman yang dapat digunakan dalam pengembangan aplikasi berbasis *mobile*, *web* maupun *desktop*. Kerangka kerja tersebut membuat pekerjaan *developer* menjadi lebih mudah dalam menuliskan sebuah program dengan efisien dan tersusun rapi.

Framework diciptakan untuk membantu para *developer* dalam mengembangkan aplikasi menjadi lebih cepat dan tersusun serta terstruktur, sehingga tidak perlu melakukan penulisan kode program secara berulang-ulang.[32]

2.2.6 Visual Studio Code

Visual studio code adalah sebuah aplikasi *code editor* yang dikembangkan oleh Microsoft dan bisa dijalankan di berbagai *platform*, karenanya *Visual Studio Code* dapat digunakan pada komputer dengan *OS Windows, Linux* maupun *MacOS*.[31]

Visual studio code merupakan *code editor* yang sangat *powerful*, handal tetapi sangat ringan ketika digunakan. Berbagai macam jenis bahasa pemrograman dapat diedit pada aplikasi ini, misalnya *TypeScript, Python, Java* hingga *Node.js* dan banyak lagi bahasa pemrograman lain yang didukung. Hal tersebut karena *Visual Studio Code* memiliki lingkungan pengembangan yang luas dan tersedianya banyak *extension* yang dapat mempermudah pekerjaan *programmer*.[31]

2.2.7 Dart

Dart merupakan bahasa pemrograman yang digunakan pada *framework Flutter*. Sebuah *framework* modern seperti *Flutter* membutuhkan sebuah bahasa pemrograman yang *high-level* untuk bisa menyediakan pengalaman yang menyenangkan bagi para *developer* dan membuat aplikasi yang berkualitas menjadi jauh lebih mudah.

Dart awalnya dikembangkan oleh Google pada 2011 dengan tujuan utamanya yaitu menggantikan *JavaScript* pada *web development*, meskipun begitu, sekarang Dart berfokus pada pengembangan aplikasi *mobile*. Dart versi stabil rilis pada 2013 dan membawa perubahan yang besar pada perilisannya di penghujung tahun 2018. Dart adalah sebuah bahasa *modern cross-platform* yang berarti bisa berjalan pada berbagai jenis sistem operasi, kemudian bahasa yang *general-purpose* yang sampai saat ini masih terus dikembangkan fitur-fiturnya yang membuatnya semakin stabil dan fleksibel. *Syntax* pada Dart mirip dengan bahasa pemrograman berorientasi objek

lainnya seperti C++, C#, *Java*, *Python*, *JavaScript* [22], selain itu Dart memiliki performa yang luar biasa, hal tersebut didukung oleh *Dart Virtual Machines* yang dapat meng-*compile* kode program ke bahasa *native* dengan 2 mode yaitu *Just In Time*(JIT) atau *Ahead-Of-Time* (AOT), dapat dijalankan pada ARM maupun kode x86 sehingga dapat berjalan pada iOS dan juga Android [23]

2.2.8 PHP

PHP adalah bahasa pemrograman yang sering digunakan dalam pengembangan suatu aplikasi web. PHP adalah singkatan dari *hypertext preprocessor* dan berjalan pada *server side scripting*. PHP juga merupakan bahasa yang bersifat *open source*. [29]

PHP merupakan bahasa *scripting server-side* yang berarti instruksi pemrogramannya akan dijalankan ketika proses *runtime* dan dijalankan serta diproses di *server*. [30]

2.2.9 MySQL

MySQL adalah sebuah *database management system* yang bersifat *open source* dan dikembangkan menggunakan *structured query language*. Banyak digunakan dalam pengembangan *website* dan aplikasi yang membutuhkan penyimpanan yang bertipe *relational database*.

Dikembangkan oleh MySQL AB asal Swedia pada tahun 1994. Kemudian diakuisisi oleh Oracle pada 2010

2.2.10 Use Case Diagram

Sebuah *use case diagram* menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem. Diagram ini bisa menjadi gambaran yang bagus untuk menjelaskan konteks dari sebuah sistem sehingga terlihat jelas batasan dari sistem (Larman, 2005). Ada 2 elemen penting yang harus digambarkan, yaitu aktor dan *use case*. Aktor adalah segala sesuatu yang berinteraksi langsung dengan sistem, bisa merupakan orang (yang ditunjukkan dengan perannya dan bukan namanya/personalnya) atau sistem komputer yang lain.

Aktor dinotasikan dengan simbol gambar orang-orangan (*stick-man*)

dengan nama kata benda di bagian bawah yang menyatakan peran/sistem. Aktor bisa bersifat primer, yaitu yang menginisiasi berjalannya sebuah *use case*, atau sekunder, yaitu yang membantu berjalannya sebuah *use case*. *Use case* dinotasikan dengan simbol elips dengan nama kata kerja aktif di bagian dalam yang menyatakan aktivitas dari perspektif aktor. Setiap aktor dimungkinkan untuk berinteraksi dengan system dalam banyak *use case*. Sebaliknya, setiap *use case* bisa dijalankan oleh lebih dari satu aktor. [25]

2.2.11 Use Case Scenario

Use case scenario merupakan penjelasan secara tekstual dari sekumpulan skenario interaksi. Setiap skenario mendeskripsikan urutan aksi/langkah yang dilakukan aktor ketika berinteraksi dengan sistem, baik yang berhasil maupun gagal. *Use case scenario* dijelaskan secara tekstual dalam

Beberapa format tergantung kebutuhannya, yaitu singkat (*brief*), informal (*casual*), atau lengkap (*fully dressed*) (Larman, 2005), yang bisa dijelaskan dalam bentuk tabel dengan 1 kolom atau 2 kolom (Cockburn, 2000). Pada format singkat, penjelasan diberikan cukup 1 paragraf yang mengacu hanya pada skenario yang berhasil. Pada format informal, penjelasan diberikan dalam beberapa paragraf yang mencakup semua skenario, baik yang berhasil maupun gagal.[26]

2.2.12 Sequence Diagram

Sequence diagram menjelaskan interaksi *object* yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use case* diagram, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *use case*. [27]

2.2.13 Entity Relational Database

Entity Relationship Diagram (ERD) adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas(*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi. [28]

2.2.14 White Box Testing

Pengujian perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan. Merawat program bisa dilakukan dengan menyederhanakan *source code* program sehingga apabila diuji menggunakan *White Box Testing* lagi, akan menghasilkan *Node*, *Edgeds* dan *Test Cases* yang lebih sedikit dibandingkan dengan pengujian sebelumnya. [22]

2.2.15 Black Box Testing

Pengujian *black box* dilakukan berdasarkan masukan dan luaran tanpa memperhatikan rincian program sehingga penguji tidak perlu memiliki pengetahuan pemrograman.[23] Pengujian *Black Box* yang memiliki arti bahwa pengujian ini hanya memeriksa suatu perangkat lunak dari hasil eksekusinya, tanpa harus tahu mengetahui kode program dan hanya memperhatikan fungsionalnya.[24] Metode pengujian *black box* terdiri dari beberapa metode, di antaranya analisis batas, pemisahan ekivalensi, pengujian perbandingan, pengujian kekokohan, dan pengujian sampel.

2.2.16 BloC_test

Dengan BloC maka setiap fungsi memiliki kode program yang terpisah, unit test hanya menjalankan *test* terpisah berdasarkan fungsionalitasnya. Artinya, kelas atau objek yang digunakan saat menjalakna pengujian unit *test* tidak boleh berdampak kepada kelas lainnya yang tidak terkait dengan *test* tersebut[10].



```

group('NamaGroupTest', () {
  blocTest(
    'Deskripsi Test',
    build: () => CounterBloc(),
    act: (register) => register.registerEvent(
      "coba1", "coba@gmail.com", "12345678"),
    expect: () => <RegisterState>[
      RegisterLoading(),
      RegisterSuccess(),
    ],
  );
});

```

Gambar 2.1 Pengujian Bloc_test

Gambar 2.1 adalah contoh pengujian menggunakan *library bloc_test*, sebuah class akan dikelompokkan pada satu grup yang sama untuk

memudahkan pada proses pengujian, kemudian setiap fungsi atau method akan diuji pada suatu fungsi tersendiri yaitu *blocTest*. *BlocTest* memiliki beberapa parameter yaitu:

- **deskripsi test**
- *build* yaitu untuk memilih *class* induk yang akan diuji,
- *act* yaitu instruksi untuk menjalankan suatu fungsi
- *expect* yaitu hasil yang diharapkan dari pengujian.