**BAB III**

**METODE PENELITIAN**

1. **Jenis Penelitian**

Jenis penelitian yang digunakan dalam penelitian ini adalah jenis penelitian *research and development* (R&D). Di mana pada penelitian ini akan menghasilkan suatu produk, yaitu aplikasi *Finding Tutor* berbasis *android*. Penelitian ini bertujuan untuk mengembangkan aplikasi dengan menggunakan pengembangan tahapan *prototyping.*

1. **Tempat dan Waktu Penelitian**

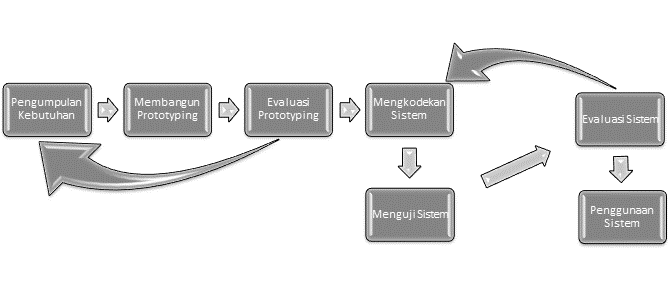
Penelitian dilaksanakan di Lembaga Anandaprivat. Jadwal pelaksanaan penelitian direncanakan pada bulan September sampai bulan November 2019.

1. **Model Pengembangan**

Perancangan aplikasi ini menggunakan model pengembangan yang digunakan adalah model pengembangan dari metode *system life cycles (*SDLC) yaitu *prototyping*. Model ini sering digunakan apabila pemilik sistem tidak terlalu menguasai sistem yang akan dikembangkannya, sehingga dia memerlukan gambaran dari sistem yang akan dikembangkannya tersebut. (Mulyani: 2016). Model pengembangan ini dipilih karena proses perancangan aplikasi ini memerlukan komunikasi yang intens antara pengembang (developer) dengan pengguna sistem.

1. **Prosedur Pengembangan**

Model pengembangan yang dipilih yaitu model *prototyping*, tahapan-tahapan dalam penelitian ini yaitu:



Gambar 3.1

Tahapan Model *Prototyping*

* + - 1. Analisis Kebutuhan
         1. Analisis Kebutuhan Media Aplikasi

Berdasarkan latar belakang yang telah dibahas sebelumnya diketahui bahwa jumlah usia remaja 474,152 (10-24) . Ini mengindikasikan banyaknya anak usia remaja membutuhkan pendalaman materi untuk meningkatkan prestasi di sekolah maupun diluar sekolah. Berdasarkan data terdapat banyak peluang bisnis, salah satunya adalah bisnis menjadi guru les atau tutor. Sementara itu, mencari guru les yang sesuai keinginan atau kebutuhan yang mana media informasi tentang guru les atau tutor tersebut yang ada di Makassar masih nihil sehingga informasi tentang guru les atau tutor itu tidak tersampaikan kepada pelajar. Untuk itu perlu dikembangkan media informasi terkait dengan guru les atau tutor yang ada di Makassar yang akan memberikan informasi guru les atau tutor, paket, waktu dan estimasi biayanya, sehingga dengan adanya pengembangan ini diharapkan pelajar maupun guru les atau tutor dapat mendapatkan informasi berdasarkan kriteria yang diinginkan.

* 1. Analisis Kebutuhan Media Aplikasi oleh Pengguna

Analisis merupakan kegiatan yang dilakukan untuk mengetahui beberapa hal terkait dengan kebutuhan pengguna dari produk yang dibuat. Bagi seorang pengembang aplikasi dalam merancang sebuah sistem hal yang pertama dilakukan yaitu melakukan analisis kebutuhan. Analisis kebutuhan tersebut bertujuan untuk mengumpulkan data yang selanjutnya dianalisis dan kemudian diolah oleh pengembang. Hasilnya akan berupa sebuah aplikasi.

Analisis kebutuhan yang dilaksanakan dalam penelitian ini menggunakan wawancara. Wawancara dilakukan untuk mengetahui fakta yang terdapat dilapangan sehingga dapat membantu pengembang dalam memahami kebutuhan pengguna yang selanjutnya akan dianalisis. Panduan dan hasil wawancara dapat dilihat pada Lampiran 3.

Menganalisis sebuah kebutuhan memerlukan ketelitian seorang pengembang. Pengembang harus memahami data yang telah dikumpulkan dan mengolahnya menjadi sebuah informasi. Selain itu, pengembang juga perlu memperhatikan beberapa hal yang menjadi penunjang dalam pengembangan aplikasinya seperti, tampilan, lingkungan sistem dan konten aplikasi serta alat dan bahan yang digunakan dalam proses pembuatan aplikasi tersebut.

* 1. Analisis Kebutuhan Perangkat Keras

1. Smarphone Minimal Spesifikasi
2. Display screen resolution 1280 x 720
3. Screen size 5.0”
4. Penyimpanan internal 16 gb
5. Memory 2 gb
6. Processor dual core 1,19Ghz
7. Leptop/Komputer Minimal Spesifikasi
8. Processor intel® core™ i5-7200U CPU @2.50Ghz (4CPUs),-2.7GHz
9. Display 14”
10. Hard disk drive 1TB\
11. Ram 4 gb
    1. Analisis Kebutuhan Perangkat Lunak
12. Smartphone minimal versi
    * + - 1. Android 5.1 (*lollipop)*
13. Leptop/komputer
14. sistem operasi windows 10
15. Microsoft word 2016
16. Microsoft visio 2016
17. Google chrome
18. Adobe XD
19. Membangun *Prototyping*

Membangun *prototyping* dengan membuat perancangan sistem sementara yang berfokus pada penyajian kepada user berdasarkan analisis kebutuhan media aplikasi pada tahap pertama. *Prototyping* yang dibangun yaitu membuat *use case, actifity diagram, flowchart* dan perancangan antar muka atau *interface*.

* 1. Arsitektur Sistem

Berikut adalah gambaran arsitektur sistem secara global dan gambaran arsitektur sistem untuk pengguna dan operator/admin.



Gambar 3.2

Arsitektur Sistem Secara Global



Gambar 3.3

Arsitektur Sistem Operator/Admin

Tahap dalam skema arsitektur sistem informasi untuk operator/admin yaitu seorang operator/admin terlebih dahulu menyiapkan data informasi mata pelajaran, pilihan paket, waktu dan jenjang pendidikan. Data tersebut yang ada pada operator/admin selanjutnya akan diinputkan ke dalam database aplikasi kemudian database tersebut dihubungkan ke dalam web server.



Gambar 3.4

Arsitektur Sistem Pengguna

Tahapan dalam skema arsitektur sistem informasi untuk pengguna yaitu seorang pengguna terlebih dahulu *merequest* mata pelajaran yang di inginkan pada aplikasi *Finding Tutor* selanjutnya aplikasi akan menerima data *request* pengguna berupa Pilihan paket, waktu, jenjang Pendidikan dan guru les. Data *request* tersebut akan diteruskan melalui internet untuk *merequest* data yang ada di *webserver*. Selanjutnya *webserver* akan memberikan *response* dari *request* sebelumnya yaitu berupa data kursus yang telah pilih untuk dilakukan pembayaran dengan cara transfer atupun *Cash On Delivery* (COD) yang akan di tampilkan di *smartphone* pengguna.

* 1. *Use Case*

Diagram *use case* digunakan untuk memodelkan proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor. Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. Secara sederhana, diagram *use case* digunakan untuk memahami fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.



Gambar 3.5

*Use Case Diagram* Pelajar



Gambar 3.6

*Use Case Diagaram* Tutor/Guru

* 1. *Activity Diagram*

*Activity diagram* berfokus pada aktifitas-aktifitas yang terjadi dan terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Berikut adalah gambar *aktifity diagram* pada aplikasi *Finding* Tutor.

* + - 1. *Activity Diagram* Pelajar
         1. *Activity Diagram* Membuka Aplikasi



Gambar 3.7

*Activity Diagram* Membuka Aplikasi

* + - * 1. *Activity Diagram* *Navigasi* Menu *Home*

|  |  |
| --- | --- |
| User | Aplikasi |
| Memilih tutor  Memilih jenjang pendidikan  Memilih waktu  Memilih paket  Memilih mata pelajaran  Menampilkan menu utama/home, terdapat mata pelajaran  Memilih navigasi home | Menampilkan order selesai  Menampilkan tutor  Menampilkan jenjang pendidikan  Menampilkan pilihan waktu  Menampilkan pilihan paket |

Gambar 3.8

*Activity Diagram* Navigasi Menu *Home* Pelajar

* + - * 1. *Actifity Diagram* Navigasi Menu Keranjang

|  |  |
| --- | --- |
| User | Aplikasi |
| Menampilkan data kerajang  Memilih navigasi keranjang |  |

Gambar 3.9

*Activity Diagram* Navigasi Menu Keranjang

* + - * 1. *Activity Diagram* Navigasi Menu Akun

|  |  |
| --- | --- |
| User | Aplikasi |
| Memilih profil, kursus yang diikuti  Menampilkan profil, kursus yang diikuti  Memilih navigasi akun | Menampilkan akun, terdapat profil, kursus yang diikuti |

Gambar 3.10

*Activity* *Diagram* Navigasi Menu Akun

* + - 1. *Activity Diagram* Tutor
      2. *Activity Diagram* Membuka Aplikasi



Gambar 3.11

*Activity Diagram* Membuka Aplikasi Tutor

* + - 1. *Activity Diagram* Navigasi Drawer Menu Profil

|  |  |
| --- | --- |
| User | Aplikasi |
| Memilih navigasi drawer Profil | Menampilkan edit Profil |

Gambar 3.12

*Activity Diagram* Navigasi *Drawer* Menu Profil

* + - 1. *Activity Diagram* Navigasi Drawer Kompetensi

|  |  |
| --- | --- |
| Guru | Aplikasi |
| Menampilkan informasi Kompetensi guru  Memilih navigasi drawer kompetensi |  |

Gambar 3.13

*Activity Diagram* Navigasi Drawer Menu Profil

* + - 1. *Activity Diagram* Navigasi Drawer Bimbingan Pelajar

|  |  |
| --- | --- |
| Guru | Aplikasi |
| Menampilkan informasi Pelajar bimbingan  Memilih navigasi drawer bimbingan pelajar |  |

Gambar 3.14

*Activity Diagram* Navigasi Drawer Drawer Bimbingan Pelajar

* + - 1. *Activity Diagram* Navigasi Drawer Deposit

|  |  |
| --- | --- |
| Guru | Aplikasi |
| Menampilkan informasi deposit guru les  Memilih navigasi drawer deposit |  |

Gambar 3.15

*Activity Diagram* Navigasi Drawer Deposit

* 1. *Flowchart*

*Flowchart* adalah alur proses dari sebuah sistem yang digambarkan secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* berfungsi untuk menganalisis alur sistematika program.



Gambar 3.16

*Flowchart* Registrasidan *Login*



Gambar 3.17

*Flowchart* Pelajar



Gambar 3.18

*Flowchart* Tutor/Guru

1. Evaluasi *Prototyping*

Evaluasi ini dilakukan oleh user apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan dan kebutuhan *user*. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulangi langkah 1,2 dan 3.

1. Mengkodekan Sistem

Dalam tahap ini hasil rancangan diterjemahkan ke dalam bahasa pemprograman yang dipilih

1. Menguji Sistem

Setelah system dikodekan dan menjadi aplikasi, selanjutnya aplikasi harus diuji untuk menentukan kelayakan apliaksi yang dikembangkan. Metode pengujian aplikasi menggunkan standar ISO 25010 *The Internasional Standar Organization* (ISO) mengembangkan standar ISO 25010 dalam upaya untuk mengindentifikasi kualitas dari perangkat lunak.

1. Evaluasi Sistem

*User* mengevaluasi apakah sistem yang sudah sesuai dengan yang diharapkan *user*. Jika iya, maka dapat dilanjutkan ke langkah 7, jika tidak, maka ulangi langkah 4 dan 5.

1. Menggunakan Sistem

Perangkat lunak yang telah diuji dan dievaluasi, perangkat lunak siap untuk digunakan.

1. **Teknik Pengumpulan Data**

Teknik pengumpulan data yang digunakan dalam penelitian ini, yaitu:

* + - 1. Studi literatur

Studi ini dikerjakan untuk mengumpulkan temuan riset dan informasi lain yang berkaitan dengan pengembangan produk yang direncanakan, yang diperoleh dari buku, hasil penelitian dan jurnal.

* + - 1. Observasi

Teknik observasi dilakukan peneliti untuk mengumpulkan data mengenai analisis kualitas perangkat lunak yang dikembangkan. Aspek kualitas yang digunakan yaitu pengujian ISO 25010 pada faktor kualitas *functionality suitability, performance efficiency, compatibility*, *usability* dan *portability.*

* + - 1. Wawancara

Pada metode ini, pengembang melakukan wawancara dengan pimpinan Anandaprivat Indonesia, sehingga didapat data-data yang dibutuhkan untuk mengembangkan aplikasi dan untuk mengevaluasi *prototyping* yang telah dibuat.

* + - 1. Kuesioner

Kuesioner merupakan teknik atau cara pengumpulan data secara tidak langsung di mana peneliti tidak langsung bertanya jawab dengan responden. Instrument atau alat pengumpulan datanya pun disebut angket yang berisi pertanyaan statement. Jenis instrument angket yang digunakan oleh peneliti adalah *checklist*. Dalam penelitian ini pihak yang terlibat yaitu lembaga anandaprivat dan pelajar sebanyak 30 responden.

1. **Instrumen penelitian**

Instrumen dalam penelitian digunakan untuk membantu dalam melakukan analisis kualitas dari perangkat lunak yang dikembangkan yaitu menggunakan *ISO* *25010* yang merupakan pengembangan dari model sebelumnya yaitu *ISO 9126*. Pada model *ISO 25010* ini terdapat delapan domain (kriteria) di mana terdiri dari sejumlah sub domain tambahan dan beberapa sub domain yang dipindahkan ke domain lainnya.

Menurut Ben David (2011:2) ada 4 aspek pengujian perangkat lunak meliputi *functional testing, compatibility testing, usability testing* dan *performance testing*. Jika dari 4 aspek menurut Ben David dibandingkan dengan pengujian ISO 25010 maka pengujian sebuah aplikasi perlu dilakukan pada karakteristik *functionality suitability, performance efficiency, usability, compatibility* dan *portability*. Oleh karena itu, dalam penelitian ini tahap pengujian perangkat lunak menggunakan 5 dari 8 karakteristik dalam ISO 25010 didasarkan dari teori Ben David. Berikut penjelasan dari 5 aspek pengujian pengujian tersebut:

* + - * 1. *Functionality Suitability*

Karakteristik ini menunjukan sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan ketika digunakan dalam kondisi tertentu. Untuk analisis aspek ini menggunakan instrumen pengujian beberapa fungsi dari aplikasi. Instrumen pengujian berupa angket yang diberikan kepada dua dosen ahli media/aplikasi untuk menilai fungsi dari aplikasi. Adapun kisi-kisi instrumen pengujiannya adalah menguji dari segi tampilan splash screen dan tampilan setiap menu.

Tabel 3.1

Kisi-kisi Instrumen *Functionality Suitability*

|  |  |  |  |
| --- | --- | --- | --- |
| Aspek | Sub-Karakteristik | Keterangan | Nomor Butir |
| *Functionality suitability* | *Functional completenesss* | Informasi dan menu yang terdapat pada aplikasi sudah lengkap | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 |
| *Functional correctness* | Tombol/menu pada aplikasi dapat digunakan | 27, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 |
| *Functional appropriatenes* | Tombol/menu yang terdapat pada aplikasi sesuai dengan kebutuhan | 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100 |

* + - * 1. *Performance Efficiency*

Karakteristik ini memungkinkan perangkat lunak memenuhi derajat penggunaan yang optimal dari sumber daya sistem dan memuat atribut-atribut *time behavior* yang merujuk pada seberapa cepat respon sistem berjalan dan *resource utilization* yang merujuk pada kemampuan perangkat lunak untuk dapat menggunakan resource dengan efisien.

Ben David (2011) mengemukakan performance testing pada pengujian aplikasi mobile adalah jenis pengujian yang menilai penggunaan memori/ CPU, konsumsi baterai, dan beban pada server dalam berbagai kondisi. Pengujian aspek *Performance efficiency* dilakukan dengan menggunakan *performance overlay* yang merupakan salah satu tools pengujian flutter. *Tools performance overlay* memiliki dua grafik yang digunakan untuk mengidentifikasi kinerja dari sebuah perangkat lunak yakni GPU dan UI(GPU) yang berjalan diatas aplikasi.

* + - * 1. *Compatibility*

Aspek ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak untuk dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama. Pengujian pada aspek *compatibility* dilakukan dengan cara mengoperasikan aplikasi *Finding* Tutor dan beberapa aplikasi pada perangkat android secara bersamaan.

Tabel 3.2

Kisi-Kisi Instrumen *Compatibility*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Aplikasi | Berhasil | Gagal |
| 1 | *Finding* Tutor berjalan bersamaan dengan *Play Store* |  |  |
| 2 | *Finding* Tutor berjalan bersamaan dengan *Facebook* |  |  |
| 3 | *Finding* Tutor berjalan bersamaan dengan *Youtube* |  |  |
| 4 | *Finding* Tutor berjalan bersamaan dengan *Maps* |  |  |
| 5 | *Finding* Tutor berjalan bersamaan dengan *Google Chrome* |  |  |

* + - * 1. *Portability*

Pengujian aspek portability dari aplikasi ini menggunakan beberapa *smartphone* dengan spesifikasi yang berbeda. *Smartphone* yang digunakan untuk pengujian *portability* aplikasi Pariwisata sulawesi barat merupakan *smartphone* *platform Android* dengan sistem operasi yaitu Lollipop (versi 5.1) hingga oreo (versi 8.0). Pada pengujian *portability*, jumlah *smartphone* yang digunakan yakni 4 *smartphone* dengan produsen yang berbeda-beda.

Tabel 3.3

Kisi-Kisi Instrumen *Portability*

|  |  |  |  |
| --- | --- | --- | --- |
| No | *Smartphone* yang digunakan | Berhasil | Gagal |
| 1 | *Lenovo a6000 +* |  |  |
| 2 | *Redmi Note 4X* |  |  |
| 3 | *Redmi 4X* |  |  |
| 4 | *Oppo A7* |  |  |

* + - * 1. *Usability*

Pengujian untuk karakteristik operability dilakukan dengan menggunakan Angket. Angket yang digunakan adalah USE Questionnaire oleh Lund A.M (2001). USE Questionnaire sudah mencakup sub karakteristik usability ISO 25010 dan pendapat Ben David pada aspek *usability testing*. Menurut Ben David pada pengujian *usability* *testing* yaitu efisien, efektif dan memuaskan untuk pengguna aplikasi. Untuk memudahkan pengguna dalam mengisi kuesioner, terlebih dahulu kuesioner diubah kedalam Bahasa Indonesia agar mudah dipahami oleh responden.

Tabel 3.4

Kisi-Kisi Instrumen Pelajar dan Tutor *Usability*

MenggunakanUSE Questionnaire oleh Lund A.M (2001)

|  |  |  |  |
| --- | --- | --- | --- |
| Aspek | Kriteria | Indikator | Nomor Butir |
| *Usability* | *Usefulness* | Aplikasi berguna sesuai dengan kebutuhan pengguna | 1, 2, 3, 4, 5, 6, 7, 8 |
| *Easy of Use* | Aplikasi mudah digunakan | 9, 10, 11, 12, 13, 14, 15, 16, 18, 19 |
| *Easy of Learning* | Aplikasi mudah dipelajari | 20, 21, 22, 23 |
| *Satisfaction* | Aplikasi memuaskan pada saat penggunaan | 24, 24, 25, 26, 27, 28, 29, 30 |

1. **Teknik Analisis Data**

Teknik analisis data yang digunakan adalah sebagai berikut:

1. Analisis Validitas Instrumen Penelitian

Instrumen yang digunakan dalam penelitian ini terlebih divalidasi oleh dua orang ahli. Pada lembar validasi instrumen penelitian jawaban setiap item pertanyaan menggunakan skala *likert*. Skala *likert* merupakan jenis skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau kelompok. Djaali (dalam helmi dkk, 2016). Adapun kriteria pemberian skor untuk *alternative* jawaban untuk setiap item sebagai berikut: (1) skor 5 untuk jawaban sangat setuju, (2) skor 4 untuk jawaban setuju, (3) skor 3 untuk jawaban kurang setuju, (4) skor 2 untuk jawaban tidak setuju, (5) skor 1 untuk jawaban sangat tidak setuju.

Tabel 3.5

Konversi Skala Likert

|  |  |
| --- | --- |
| Jawaban | Skor |
| Sangat Tidak Setuju | 1 |
| Tidak Setuju | 2 |
| Kurang Setuju | 3 |
| Setuju | 4 |
| Sangat Setuju | 5 |

Berikut ini merupakan rumus dari perhitungan yang dapat digunakan untuk mengolah data kuesioner.

Persentase kelayakan (%) =

Setalah mendapatkan data skor dari hasil pengujian, kemudian dihitung presentasenya dengan menggunakan rumus tersebut. Setelah itu, persentase dikonversi ke dalam pernyataan sesuai dengan tabel berikut. Sudaryono (dalam jayanto, 2017).

Tabel 3.6

Interpretasi Presentase Kelayakan

|  |  |
| --- | --- |
| Persentase Kelayakan | Kriteria |
| 0% - 20% | Sangat Tidak Layak |
| 21% - 40% | Kurang Layak |
| 41% - 60% | Cukup Layak |
| 61% - 80% | Layak |
| 81% - 100% | Sangat Layak |

1. Analisis Validitas Konten

Analisis untuk validitas konten dilakukan dengan teknik deskriptif yaitu menganalisis persentase instrumenoleh lembaga anandaprivat dan hasil perhitungan skor persentase untuk masing-masing penilaian. Pada lembar validasi ahli media jawaban setiap item menggunakan skala Guttam. Menurut Sugiyono (2010), skala pengukuran dengan tipe ini akan didapat jawaban yang tegas yaitu ya-tidak, benar-salah, pernah-tidak pernah, positif-negatif. Jawaban dapat dibuat dalam bentuk *checklist* dengan skor tinggi satu dan skor rendah nol.

Tabel 3.7

Kategori Pemberian Skor Alternatif Jawaban

|  |  |
| --- | --- |
| Jawaban | Skor |
| Ya | 1 |
| Tidak | 0 |

Sumber: (Sugiyono, 2016 :96)

Hasil pengujian dihitung dengan rumus dari matriks *Feature Completeness*. Matriks *feature completeness* merupakan matrik untuk mengukur sejauh mana fitur-fitur dapat diimplementasikan dengan benar. Berikut ini merupakan rumus dari perhitungan yang dapat digunakan untuk mengolah data kuesioner.

*I* = Jumlah fungsi yang dirancang

*P* = Jumlah fungsi yang berhasil diimplementasikan

Dalam matriks *Feature Completeness*, nilai yang mendekati 1 mengindikasikan banyaknya fitur yang berhasil diimplementasikan. Hasil diukur dalam skala . Perangkat lunak dikatakan baik dalam *functional suitability* jika X mendekati nilai 1.

1. Analisis *Functional Suitability*

Analisis untuk aspek *fungcional suitability* dilakukan dengan teknik deskriptif yaitu menganalisis persentase *fungcional* oleh ahli media dan hasil perhitungan skor persentase untuk masing-masing penilaian. Pada lembar validasi ahli media jawaban setiap item menggunakan skala Guttamn. Menurut Sugiyono (2010), skala pengukuran dengan tipe ini akan didapat jawaban yang tegas yaitu ya-tidak, benar-salah, pernah-tidak pernah, positif-negatif. Jawaban dapat dibuat dalam bentuk *checklist* dengan skor tinggi satu dan skor rendah nol.

Tabel 3.8

Kategori Pemberian Skor Alternatif Jawaban

|  |  |
| --- | --- |
| Jawaban | Skor |
| Ya | 1 |
| Tidak | 0 |

Sumber: (Sugiyono, 2016 :96)

Hasil pengujian dihitung dengan rumus dari matriks *Feature Completeness*. Matriks *feature completeness* merupakan matrik untuk mengukur sejauh mana fitur-fitur dapat diimplementasikan dengan benar. Berikut ini merupakan rumus dari perhitungan yang dapat digunakan untuk mengolah data kuesioner.

*I* = Jumlah fungsi yang dirancang

*P* = Jumlah fungsi yang berhasil diimplementasikan

Dalam matriks *Feature Completeness*, nilai yang mendekati 1 mengindikasikan banyaknya fitur yang berhasil diimplementasikan. Hasil diukur dalam skala . Perangkat lunak dikatakan baik dalam *functional suitability* jika X mendekati nilai 1.

1. Analisis *Performance Efficiency*

Analisis kualitas aspek *performance efficiency* dilakukan dengan menguji performa aplikasi saat dioperasikan dalam sebuah *smartphone*. Untuk menguji performa aplikasi digunakan unit analisis *tools* *performance overlay*. Pengujian berfokus pada penggunaan GPU dan UI(CPU). Di dalam *tools* *performance overlay* penggunaan avg GPU dan UI(CPU) dibawah 16ms/frame dan indikator garis berwarna hijau dikatakan normal.

1. Analisis *Usability*

Pengujian *Usability* diujicobakan pada 30 responden yang terdiri dari pelajar dan tutor, menggunakan skala *Likert* dengan 5 skala rincian.

Tabel 3.9

Konversi Skala Likert

|  |  |  |
| --- | --- | --- |
| Penilaian | Keterangan | Skor |
| STB | Sangat Tidak Baik | 1 |
| TB | Tidak Baik | 2 |
| CB | Cukup Baik | 3 |
| B | Baik | 4 |
| SB | Sangat Baik | 5 |

Berikut ini merupakan rumus dari perhitungan yang dapat digunakan untuk mengolah data kuesioner.

Persentase kelayakan (%) = ..............3.1

Setalah mendapatkan data skor dari hasil pengujian, kemudian dihitung presentasenya dengan menggunakan rumus tersebut. Setelah itu, persentase dikonversi ke dalam pernyataan sesuai dengan tabel berikut. Sudaryono (dalam jayanto, 2017).

Tabel 3.40

Konversi Kualitatif dari Persentase Kelayakan

|  |  |
| --- | --- |
| Persentase% | Kriteria |
| 81 – 100 | Sangat Baik |
| 61 – 80 | Baik |
| 41 – 60 | Cukup Baik |
| 21 – 40 | Kurang Baik |
| 0 – 20 | Sangat Tidak Baik |

1. Analisis *Compatibility* dan *Portability*

Pengujian aspek *compability* dan *portability* dilakukan dengan melakukan observasi menggunakan skala gutmam, observasi dilakukan dengan cara menjalankan beberapa aplikasi secara bersamaan dan menjalankan aplikasi pada beberapa *smartphone* yang berbeda. Analisis *compatibility* dan *portability* dilakukan untuk mengetahui apakah aplikasi Anandaprivat dapat melakukan fungsi yang dibutuhkan ketika dijalankan bersamaan dengan aplikasi yang lain pada perangkat yang sama dan pada perangkat yang berbeda. Pada lembar observasi jawaban setiap item menggunakan skala *guttman*. Skala *guttman* dikembangkan oleh Louis Guttman. skala ini digunakan untuk mendapatkan jawaban yang tegas, contohnya “ya” atau “tidak”, “pernah” atau “tidak pernah”, “positif” atau “negative”, “benar” atau “salah” dan lain-lain. Jawaban dibuat dalam bentuk *checklist* dengan skor tinggi satu dan rendah nol.

Tabel 3.41

Konversi Skala Guttman

|  |  |
| --- | --- |
| Jawaban | Skor |
| Ya | 1 |
| Tidak | 0 |

Sumber: (Sugiyono, 2016 :96)

Hasil pengujian dihitung dengan rumus dari matriks *Feature Completeness*. Matriks *feature completeness* merupakan matrik untuk mengukur sejauh mana fitur-fitur dapat diimplementasikan dengan benar. Berikut ini merupakan rumus dari perhitungan yang dapat digunakan untuk mengolah data kuesioner.

*I* = Jumlah fungsi yang dirancang

*P* = Jumlah fungsi yang berhasil diimplementasikan

Dalam matriks *Feature Completeness*, nilai yang mendekati 1 mengindikasikan banyaknya fitur yang berhasil diimplementasikan. Hasil diukur dalam skala . Perangkat lunak dikatakan baik dalam *functional suitability* jika X mendekati nilai 1.