

BAB IV

METODE PENELITIAN

4.1 Jenis Data

4.1.1 Data Primer

Data primer adalah data yang diperoleh atau dikumpulkan oleh orang yang melakukan penelitian atau yang bersangkutan yang memerlukannya. (Hasan, 2003:33). Dalam penulisan ini data primer didapat oleh penulis melalui kegiatan mengamati *game* media pembelajaran yang sudah ada seperti *Game Matematika Anak*, *Tebak Gambar*, *Mental Math*, *Kids Number and Math Lite*, *Penguin Jump* dan *Save the Hamster*. data yang diamati seperti bagaimana mengikuti pembelajaran, layout game, button navigasi dan skor.

4.1.2 Data Sekunder

Data sekunder adalah data yang diperoleh atau dikumpulkan dari sumber-sumber yang telah ada. (Hasan, 2003:33). Dalam penulisan ini data sekunder didapat oleh penulis dari laporan-laporan penelitian terdahulu, penulis juga menggunakan referensi buku dan jurnal-jurnal penelitian.

4.2 Teknik Pengumpulan Data

4.2.1 Observasi

Menurut *Indrajani (2011:5)*, *observasi* adalah salah satu teknik pencarian data paling efektif untuk pemahaman suatu sistem. Menurut *Jogiyanto (2005:623)*, *observasi* adalah pengamatan langsung suatu kegiatan yang sedang dilakukan.

Teknik *observasi* dilakukan bersama-sama dengan pengumpulan kebutuhan sistem yang lain. Penulis mengamati game edukasi seperti *Game Matematika Anak*, *Tebak Gambar*, *Mental Math*, *Kids Number and Math Lite*, *Penguin Jump* dan *Save the Hamster*. Adapun yang penulis amati adalah:

1. Menu yang mudah dimengerti
2. Tampilan atau desain game yang menarik
3. *Backsound game*
4. *Tutorial game*
5. Nilai atau *score game* untuk user

4.2.2 Studi Pustaka

Menurut *Indrajani (2011:1)*, studi pustaka merupakan metode yang dilakukan dengan cara mengumpulkan, membaca, dan

mempelajari data-data yang ada dari berbagai media, seperti buku-buku, hasil karya tulis, jurnal-jurnal penelitian atau artikel-artikel dari internet yang berhubungan

dengan masalah yang dibahas. Dalam hal ini penulis mengumpulkan data dari buku dan jurnal pengembangan multimedia.

4.3 Alat dan Pengembangan Sistem

4.3.1 Alat Pengembangan Sistem

Menurut Rosa dan Shalahuddin (2014:137), *Unified Modeling Language (UML)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi

penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

4.3.1.1. Model Proses

a. Use Case Diagram

Menurut Rosa dan Shalahuddin (2014:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan


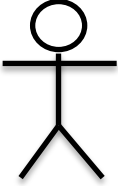

dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

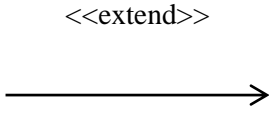

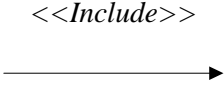
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua ahal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut simbol-simbol *use case* menurut (Rosa dan Shalhuddin, 2014: 156):

Tabel 4.1 Simbol – Simbol *Use Case*

Nama Komponen	Simbol	Keterangan
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>
<i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i> .
<i>Association</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi pada aktor.

Nama Komponen	Simbol	Keterangan
Ektensi / <i>extend</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama dapan yang sama dengan <i>use case</i> yang tambahkan.
Generalisasi / <i>generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber: Rosa dan Shalahuddin (156:2014)

b. Class Diagram

Menurut Rosa dan Shalahuddin (2014:141), Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi

pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas

sebaiknya memiliki jenis-jenis kelas berikut ini :

1. Kelas main
Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai
3. Kelas yang diambil dari pendahuluan use case Kelas yang menangani fungsi-fungsi yang harus ada di ambil dari pendefinisian *use case*.
4. Kelas yang diambil dari pendefinisian data
Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada.

c. *Activity Diagram*



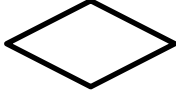


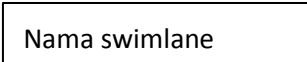
Menurut Rosa dan Shalahuddin (2014: 161), Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal – hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokkan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antaruka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

4. Rancangan menu yang ditampilkan pada perangkat.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 4.2. Simbol Activity Diagram

Nama Komponen	Simbol	Keterangan
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / decision		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / join		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Rosa dan Shalahuddin (162:2014)

4.3.2 Teknik Pengembangan Sistem

Teknik pengembangan multimedia yang digunakan oleh penulis menggunakan Metode Pengembangan Multimedia Development Life Cycle terdiri dari enam tahap, yaitu *concept*, *design*, *collecting content material*, *assembly*, *testing*, dan *distribution*. Berikut penjelasannya:

1. *Concept*

Tahap *concept* (konsep) adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi audience). Selain itu menentukan macam aplikasi (presentasi, interaktif, dll) dan tujuan aplikasi (hiburan, pelatihan, pembelajaran, dll).

2. *Design*

Design (perancangan) adalah tahap membuat spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan material/bahan untuk program.

3. *Material Collecting*

Material Collecting adalah tahap dimana pengumpulan bahan yang sesuai dengan kebutuhan dilakukan. Tahap ini dapat dikerjakan paralel dengan tahap *assembly*. Pada beberapa kasus,

tahap *Material Collecting* dan tahap *Assembly* akan dikerjakan secara linear tidak paralel.

4. *Assembly*

Tahap *assembly* (pembuatan) adalah tahap dimana semua objek atau bahan multimedia dibuat. Pembuatan aplikasi didasarkan pada tahap design.

5. *Testing*

Dilakukan setelah selesai tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan dilihat apakah ada kesalahan atau tidak. Tahap ini disebut juga sebagai tahap pengujian alpha (*alpha test*) dimana pengujian dilakukan oleh pembuat atau lingkungan pembuatnya sendiri.

6. *Distribution*

Tahapan dimana aplikasi disimpan dalam suatu media penyimpanan. Pada tahap ini jika media penyimpanan tidak cukup untuk menampung aplikasinya, maka dilakukan kompresi terhadap aplikasi tersebut.

4.4 Alat dan Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak sebagai suatu elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian

pokok dari spesifikasi, desain, dan pengkodean. Adapun teknik yang digunakan oleh penulis adalah pengujian *Black-Box*. Pengujian *Black-Box* berfokus pada penyerahan fungsional perangkat lunak dengan demikian pengujian *Black-Box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya dengan menggunakan semua persyaratan fungsional untuk satu program. Pengujian *Black-Box* merupakan alternatif dari teknik *White Box*, tetapi merupakan pendekan komplementer yang kemungkinan besar mampu mengungkapkan kelas kesalahan dari pada metode *White Box*.

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

- a. Fungsi-fungsi yang tidak benar atau hilang.
- b. Kesalahan *interface*.
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan lahan kinerja.
- e. Inisialisasi dan kesalahan terminasi.

Pengujian *Black-Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pegujian *Black-Box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi

perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.