

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan:

1. RGP & Hadi (2019) dengan judul “Rancang Bangun Aplikasi Belajar Pemrograman Dengan *Game Education* Pada *Smartphone* Berbasis Android” pada Jurnal Vokasional Teknik Elektronika dan Informatika (VOTEKNIKA). Penelitian ini menghasilkan menghasilkan media pembelajaran yang dapat membantu guru dan siswa dalam proses pembelajaran dengan memuat *game education* yang dapat menarik minat siswa dalam belajar
2. Pradana et al, 2018 dengan judul “Pengembangan Aplikasi Pembelajaran Pemrograman *Java* Yang Aktraktif Berbasis Android” pada Jurnal Pengembangan Teknologi Infromasi dan Ilmu Komputer. Penelitian ini menghasilkan sebuah aplikasi untuk meningkatkan minat pengguna dalam belajar bahasa pemrograman *Java* sehingga dapat menunjang proses perkuliahan mahasiswa khususnya pada mata kuliah pemrograman dasar dan juga seluruh pengguna secara umum.
3. Rahman and Haryanto, (2020) dengan judul “Perancangan Media Pembelajaran Metamorfosis Serangga Menggunakan *Augmented Reality* Berbasis Android” Penelitian ini menghasilkan sebuah aplikasi pembelajaran

dengan memanfaatkan teknologi *augmented reality* untuk menampilkan objek 3 dimensi sebagai media belajar.

4. Mubaraq, Kurniawan, and Saleh 2018 dengan judul “Implementasi *Augmented Reality* Pada Media Pembelajaran Buah-buahan Berbasis Android” pada *IT Journal*. Penelitian ini menghasilkan sebuah aplikasi media pembelajaran yang digunakan untuk proses belajar mengajar dengan menampilkan gambar 3 dimensi guna menarik minat belajar anak usia dini serta memberikan pengalaman yang berbeda dalam menerima materi pengenalan buah.
5. Sianturi & Yusfrizal, 2020 dengan judul “Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode *Reverse Cipher* Dan Rsa Berbasis Android” pada *Jurnal Teknik Informatika Kaputama (JTik)*. Penelitian ini menghasilkan sebuah aplikasi yang dirancang dan dibangun sebuah untuk pengamanan sebuah data yang akan dikirim melalui *smartphone* dengan metode *Reverse Cipher* yang dikombinasikan dengan metode *RSA* agar pengamanan data menjadi lebih baik dengan tingkat kriptografi yang baik.

II.2. Pengertian Rancang Bangun

Menurut (Christian, Hesinto, and Agustina 2018), rancang bangun adalah program yang menentukan aktifitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai atau pengguna komputer.

Sedangkan menurut (Ferdiansyah 2018), rancang bangun adalah suatu kegiatan menerjemahkan hasil analisa ke dalam bentuk perangkat lunak ada kemudian menciptakan sistem ataupun memperbaiki sistem yang sudah ada.

Dengan demikian, penulis menyimpulkan pengertian dari rancang bangun untuk menyelesaikan tugas-tugas khusus yang berasal dari pemakai atau pengguna.

II.3. Aplikasi

Menurut (Listianto et al. 2017), aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer

Sedangkan menurut (Siregar, Siregar, and Melani 2018), aplikasi merupakan penerapan, menyimpan suatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk diterapkan menjadi sebuah bentuk yang baru.

Dengan demikian, penulis menyimpulkan pengertian aplikasi adalah perangkat lunak komputer yang digunakan untuk user dalam hal menyelesaikan masalah khusus yang akan diterapkan menjadi sebuah bentuk yang baru.

II.4. Pembelajaran

Pembelajaran adalah sebuah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Pembelajaran merupakan bantuan yang diberikan pendidik agar terjadi pemerolehan ilmu dan pengetahuan, penguasaan kemahiran dan tabiat, serta pembentukan sikap dan kepercayaan pada peserta didik (Bakti et al. 2016).

Pembelajaran dengan menggunakan media pembelajaran dengan bentuk aplikasi android pada tahap orientasi belajar dan mengajar akan sangat membantu

tingkat efektifitas proses pembelajaran dan penyampaian pesan dan isi pelajaran pada saat itu(Alfagiar, Rosnelly, and Wahyuni 2020).

Dengan demikian, penulis menyimpulkan pembelajaran adalah proses interaksi peserta didik dengan pendidik yang akan menumbuhkan dan mendorong peserta didik melakukan proses belajar dan dengan diterapkannya kedalam bentuk aplikasi akan sangat membantu proses belajar mengajar.

II.5. Bahasa Pemrograman

Menurut (Saptarini, Hidayat, and Ciptayani 2019), bahasa pemrograman merupakan sebuah alat komunikasi antara manusia dengan perangkat komputer. Pembelajaran bahasa pemrograman dapat dilakukan melalui pendidikan formal maupun informal seperti lembaga kursus atau pembelajaran secara *online*.

Sedangkan menurut (Zefriyenni and Santoso 2015), bahasa pemrograman adalah suatu alat komunikasi standar untuk mengekspresikan instruksi kepada komputer. Layaknya bahasa manusia, bahasa itu juga memiliki tata tulis (*syntax*) dan aturan tertentu, bahasa pemrograman memfasilitasi cara dan aturan yang dilakukan oleh *programmer* untuk menuliskan perintah menyimpan, mengkompilasi dan melihat hasil secara benar.

Dengan demikian, penulis menyimpulkan pengertian bahasa pemrograman adalah alat komunikasi antara manusia dengan komputer untuk mengekspresikan instruksi kepada komputer.

II.6. Java

Java adalah sebuah bahasa pemrograman *scripting* yang sering digunakan dalam pembuatan aplikasi berbasis *handphone* dan juga dapat digunakan untuk menyediakan akses sebuah objek yang akan disisipkan di aplikasi. *Java* juga berfungsi sebagai penambah tingkah laku agar *widget* dapat tampil lebih atraktif (Sallaby, Utami, and Arliando 2015).

Sedangkan menurut Ali Mahrus, 2018), *Java* adalah sebuah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *Java* ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *JRE (Java Runtime Envirotment)*. *JDK* wajib terinstal pada komputer yang akan melakukan proses pembuatan aplikasi berbasis *Java*, seperti android.

Dengan demikian, penulis menyimpulkan bahwa *Java* adalah bahasa pemrograman yang digunakan untuk membuat aplikasi yang biasa digunakan untuk perangkat *mobile*.



Gambar II.1 Logo Java

Sumber: Wikipedia

II.7. Android

Android adalah sistem operasi (*Operating System*) yang umumnya digunakan pada perangkat dengan navigasi *full touch screen* yang biasa dimiliki oleh *smartphone* dan komputer tablet. Android sudah diambil alih oleh perusahaan *Google Inc* yang telah membelinya pada tahun 2005 dari *Android Inc*. Google menyediakan software/tools yang dikembangkan khusus untuk dijadikan alat pengembang aplikasi android (Yusfrizal 2019).



Gambar II.2 Logo Android

Sumber: (Hendra, dkk, 2015)

II.7.1. Versi Android

Android memiliki versi-versi sebagai berikut :

1. Android 1.5 *Cupcake*

Cupcake dirilis 30 April 2009. *Cupcake* menjadi versi android pertama yang menggunakan nama makanan. Dahulu katanya versi ini seharusnya versi 1.2, namun *Google* memutuskan untuk membuat revisi besar dan membuatnya menjadi versi 1.5 *Cupcake* adalah kue kecil yang dipanggang dalam cetakan berbentuk *cup* (Lengkong, Sinsuw, and Lumenta 2015).

2. Android 1.6 *Donut*

Android V1.6, *codename Donut*, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan *reboot*, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin, bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan topping di atasnya (Lengkong et al. 2015).

3. Android 2.0/2.1 *Éclair*

Android 2.0/2.1 *Eclair* dirilis 26 Oktober 2009. *Eclair* adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya (Lengkong et al. 2015).

4. Android 2.2 *Froyo*

Dirilis 20 Mei 2010, menggunakan *codename Froyo*, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari *Yoghurt*. *Froyo* singkatan dari *Frozen Yoghurt*, *Froyo* adalah *yoghurt* yang telah mengalami proses pendinginan, sehingga secara terlihat sama seperti es krim (Lengkong et al. 2015).

5. Android 2.3 *Gingerbread*

Android versi 2.3 *Gingerbread* dirilis resmi tanggal 6 Desember 2010. *Gingerbread* merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia (Lengkong et al. 2015).

6. Android 3.0 *Honeycomb*

Dirilis tanggal 22 February 2011. *Honeycomb* adalah sereal sarapan manis yang sudah dibuat oleh Posting Sereal. Seperti namanya, *Honeycomb*/sarang lebah,

sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu (Lengkong et al. 2015).

7. Android 4.0 *Ice Cream Sandwich*

Android 4.0-4.0.2 API Level 14 dan 4.0.3 API Level 15 pertama dirilis 19 Oktober 2011 dan dinamai *Ice Cream Sandwich*. *Ice Cream Sandwich* adalah lapisan es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi Panjang (Lengkong et al. 2015).

8. Android 4.1 *Jelly bean*

Android *Jelly Bean* diluncurkan pertama kali pada Juli 2012, dengan berbasis *Linux Kernel* dari Android 4.1 API Level 16, Android 4.2 API Level 17, Android 4.3 API Level 18. Penamaan mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit (Lengkong et al. 2015).

9. Android 4.4 *KitKat*

Android 4.4 *Kitkat* API level 19. *Google* mengumumkan Android *KitKat* (dinamai dengan izin *Nestle* dan *Hershey*) pada 3 september 2013, dengan tanggal rilis 31 Oktober 2013. *KitKat* merupakan merek sebuah coklat yang dikeluarkan oleh *Nestle*. (Lengkong et al. 2015).

10. Android 5.0 *Lollipop*

Android 5.0 *Lollipop* adalah versi stabil terbaru dari sistem operasi Android yang dikembangkan oleh *Google*, yang pada saat ini mencakup versi antara 5.0 dan 5.1.

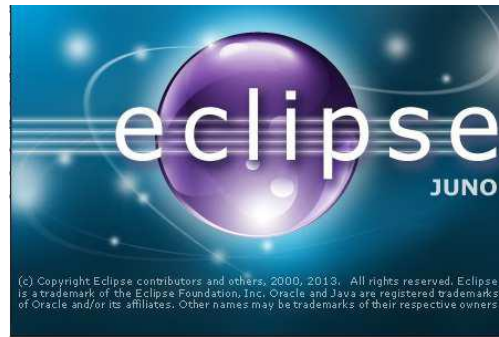


Gambar II.3 Logo Android 5.0 Lollipop

Sumber: <http://milandikatech.blogspot.com>

II.8. Eclipse

Eclipse adalah sebuah IDE (IntegratedDevelopmentEnvironment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Eclipse merupakan komunitas opensource yang bertujuan menghasilkan platform pemrograman terbuka Eclipse memiliki beberapa kelebihan yang membuatnya banyak digunakan dalam pengembangan perangkat lunak diantaranya bisa dijalankan berbagai sistem operasi seperti Windows, Linux, Solasris, Mac, dll. Dikembangkan dengan bahasa Java, namun Eclipse juga mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C++, Phyton, PHP, dan lain-lain, yang membuat Eclipse disebut juga multy-language. Multy – role, selain sebagai IDE, Eclipse juga bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dll (Dewi et al. 2018).



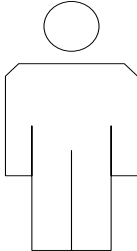
Gambar II.4 Tampilan Splash Screen *Eclipse Juno*

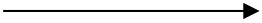
Sumber: (Hendra, dkk, 2015)

II.9.1. *Use Case Diagram*

Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat, *Use Case* digunakan untuk mengetahui fungsi yang ada didalam sistem informasi tersebut (Hariman and Puspasari 2020). Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel II.1 Simbol-Simbol *Use Case Diagram*

Simbol	Deskripsi
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja
	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat

	bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

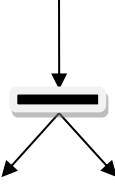
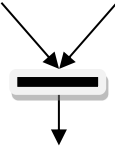
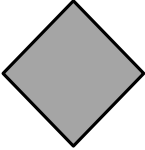
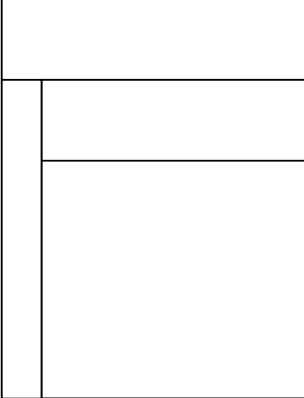
Sumber: (Hendini 2016)

II.9.2. Activity Diagram

Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam activity Diagram yaitu:

Tabel II.2 Simbol-Simbol Activity Diagram

Simbol	Deskripsi
	Start Point, diletakkan pada pojok kiri atas dan merupakan awal aktivitas

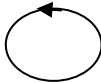
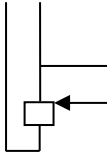
	End Point, akhir aktivitas
	Activities, menggambar kan suatu proses/kegiatan bisnis
	<i>Fork</i> /percabangan,digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Join (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>
	Swimlane, pembagian activity diagram untuk menunjukkan siapa melakukan apa

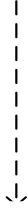
Sumber: (Hendini, 2016)

II.9.3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel II.3 Simbol-Simbol *Sequence Diagram*

Simbol	Deskripsi
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

Sumber: (Hendini 2016)

II.9.4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi: Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

Tabel II.4 Simbol-Simbol *Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

Sumber: (Hendini 2016)