

BAB II

LANDASAN TEORI

2.1 Aplikasi *Mobile*

(Prakarsya, 2019) menyatakan bahwa, *mobile* adalah istilah yang digunakan untuk menggambarkan aplikasi pada perangkat kecil, portabel, dan nirkabel yang mendukung komunikasi. Kata *mobile* memiliki arti bergerak atau pindah, jadi aplikasi *mobile* adalah sebutan untuk aplikasi yang berjalan pada perangkat bergerak. Dengan aplikasi *mobile*, Anda dapat dengan mudah melakukan berbagai aktivitas lainnya.

2.2 *Android*

(Firly, 2018) menyatakan bahwa *Android* merupakan sistem operasi untuk *smartphone* dan *tablet*. Sistem operasi dapat digambarkan sebagai jembatan antara perangkat (*device*) dengan penggunanya, sehingga pengguna dapat berinteraksi dengan perangkatnya dan menjalankan aplikasi yang tersedia pada perangkat tersebut. *Android* dapat dikatakan *open source* sehingga siapapun dapat berpartisipasi dalam mengembangkan aplikasi.

Android pertama kali dirilis pada Oktober 2003 oleh Andy Rubin, Rich Miner, Nich Sears dan Chris White di bawah sebuah perusahaan bernama *Android Inc.* Di Palo Alto, California dan akhirnya *Android* diakuisisi oleh Google pada tahun 2005.

2.3 Perangkat Lunak Pengembangan Sistem

Beberapa *software* yang digunakan untuk mendukung pengembangan aplikasi pemesanan *mobile* berbasis *android* untuk aplikasi tersebut diantaranya yaitu.

2.3.1 *Android Studio*

(Firly, 2018) menyatakan bahwa *Android studio* merupakan *integrated development environment (IDE)* atau dalam arti lain merupakan *integrated development environment* yang dirancang khusus untuk pengembangan sistem operasi *Google Android*. Aplikasi ini dibangun di atas perangkat lunak *IntelliJ IDEA Seamless JetBrains*. Bisa juga dikatakan bahwa *android studio* merupakan pengganti dari *Eclipse android development tool* atau *ADT*

sebagai *IDE* utama dalam pengembangan aplikasi *native android*. *Android studio* diluncurkan pada 16 Mei 2013 dalam konferensi *google I/O* yang saat itu masih dalam tahap akses versi 0.1 sebagai pionir. Hingga akhirnya versi stabil 3.0 dirilis pada pertengahan Oktober 2017 dan menjadi *software* terlaris di kalangan *developer* muda. Aplikasi ini dapat digunakan di berbagai sistem operasi yaitu *Windows*, *Linux* dan *macOS*. Aplikasi ini menawarkan bermacam fitur canggih yang akan meningkatkan kemampuan produktivitas dalam proses pengembangan aplikasi. Berikut ini beberapa hal yang akhirnya mengundang banyak *developer* untuk melirik *Android Studio* sebagai *software developer*:

- a. Dukungan dari *C++*, *NDK* dan sekarang *kotlin*.
- b. Perkembangan yang *up to date*.
- c. Sistem berbasis *Gradle* yang dinilai fleksibel.
- d. Lingkungan yang mencakup seluruh perangkat *android*.
- e. *Emulator* yang cepat dan kaya akan fitur.
- f. Alat pengujian dan kerangka yang juga *ekstensif*
- g. *Instant Run*
- h. Dukungan *google cloud platform*.

2.3.2 Java

(Sibarani et al., 2018) menyatakan bahwa *Java* adalah bahasa pemrograman populer, yang dikembangkan oleh Sun Microsystems. Bahasa pemrograman ini bersifat multiplatform, artinya dapat digunakan di berbagai platform, seperti desktop, *android*, bahkan untuk sistem operasi *Linux*. Elemen dasar dari pemrograman *java* adalah pengidentifikasi, pemisah, kata kunci, dan tipe data, yang dibagi menjadi 2 kategori termasuk tipe kelas dan tipe primitif.

2.3.3 Mysql

(Madcoms, 2016) menyatakan bahwa *MySQL* adalah sistem manajemen database *SQL* yang bersifat Open Source dan paling populer saat ini. Sistem database *MySQL* mendukung beberapa fitur seperti *multithreaded*, *multi-user* dan *SQL* database manajemen system (*DBMS*). Database ini biasanya dibuat untuk keperluan suatu sistem database yang dimana prosesnya cepat, handal

dan mudah digunakan. *MySQL* menggunakan sistem tabel, kolom dan baris. Jadi wajar programmer biasanya menggunakan database *MySQL* dikarenakan cara penggunaannya yang umum digunakan banyak orang.

2.3.4 Adobe XD

Adobe XD adalah perangkat lunak desain berbasis vektor untuk aplikasi seluler dan *web*. *Adobe XD* secara resmi diluncurkan pada 14 Maret 2016, menawarkan berbagai fitur dan alat responsif yang lebih akrab. Dengan adanya *Adobe XD*, para desainer desain aplikasi *mobile* tidak kesulitan dalam mengerjakan pekerjaannya. Perangkat lunak ini mendukung situs *web wireframing* dan pembuatan prototipenya. *Adobe* menyediakan penawaran gratis *Adobe XD* untuk membuat *UI*, *UX*, dan membuat prototipe ke berbagai platform termasuk *web*, seluler, *tablet*, dan sejenisnya. (dianisa.com/pengertian-adobe-xd/)

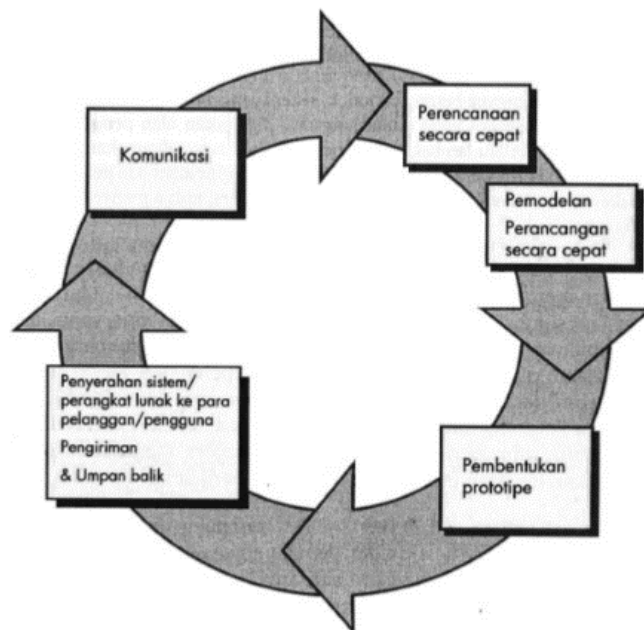
2.4 Selection Sort

(E, Setyaningsih., 2012) menyatakan bahwa *Selection Sort* merupakan kombinasi antara *sorting* dan *searching*. Untuk setiap proses, akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil atau terbesar akan dipertukarkan ke posisi yang tepat di dalam array. Untuk putaran pertama, akan dicari data dengan nilai terkecil dan data ini akan ditempatkan di indeks terkecil (*data[1]*), pada putaran kedua akan dicari data kedua terkecil, dan akan ditempatkan di indeks kedua (*data[2]*). Teknik pengurutan dgn cara pemilihan elemen atau proses kerja dgn memilih elemen data terkecil untuk kemudian dibandingkan & ditukarkan dgn elemen pada data awal, dan seterusnya sampai dengan seluruh elemen sehingga akan menghasilkan pola data yg telah *disort*. Perhitungan algoritma ini digunakan untuk mengurutkan data secara *ascending* maupun *discending*, dari nilai yang rendah ke nilai yang lebih tinggi atau sebaliknya. Jika ada *N* data dan mulai dari data ke 0 sampai *N-1* maka proses pengurutan datanya adalah sebagai berikut:

1. Mencari data minimum pada urutan data $j=0$ sampai $N-1$
2. Jika ditemukan data minimum, lakukan pertukaran posisi dengan data di posisi i
3. Proses 1 dan 2 dilakukan berulang-ulang dengan $j=j+1$ sampai $j=N-1$

2.5 Metode Pengembangan Perangkat Lunak

(Pressman, 2012) menyatakan bahwa meskipun *prototyping* dapat digunakan sebagai model yang berdiri sendiri, *prototyping* lebih umum sebagai teknik yang dapat diimplementasikan dalam konteks model proses perangkat lunak, dan paradigma *prototyping* sering membantu tim pengembangan perangkat lunak dan pemangku kepentingan lebih memahami apa yang akan dilakukan. dikembangkan ketika spesifikasi persyaratan tidak jelas.



Gambar 2. 1 Tahapan Pengembangan Aplikasi

Berikut ini adalah penjelasan dari gambar 2.1

1. Komunikasi

Tahap komunikasi ini adalah tahapan komunikasi antara *developer* dan pelanggan mengenai tujuan pembuatan *software*, mengidentifikasi sasaran keseluruhan untuk perangkat lunak yang akan di buat.

2. Perancangan secara cepat

Tahap perancangan secara cepat ini adalah tahapan perancangan cepat setelah terjalin komunikasi.

3. Pemodelan rancangan secara cepat

Tahap pemodelan perancangan secara cepat ini merupakan tahap membuat model, dan pemodelan cepat berfokus pada penggambaran seperti apa perangkat lunak apakah *visible* menurut pelanggan.

4. Pembentukan prototipe

Tahap pembentukan prototipe ini merupakan tahapan pemodelan cepat memandu pada pembuatan dari prototipe.

5. Penyerahan sistem/perangkat lunak ke para pelanggan/pengguna pengiriman dan umpan balik.

Tahap penyerahan sistem / perangkat lunak ke para pelanggan / pengguna pengiriman dan umpan balik tahapan ini adalah prototipe yang dikirimkan kemudian dievaluasi oleh pelanggan untuk memperhalus spesifikasi kebutuhan.

2.6 Pengujian Kotak Hitam

(Pressman, 2012) menyatakan bahwa pengujian kotak hitam, juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Maksudnya, teknik pengujian kotak hitam memungkinkan untuk membuat beberapa set kondisi *input* yang akan sepenuhnya melakukan semua persyaratan fungsional untuk program. Pengujian kotak hitam mencoba menemukan kesalahan dalam kategori berikut:

1. Fungsi yang salah atau hilang,
2. Kesalahan antarmuka,

3. Kesalahan dalam struktur data atau akses basis data eksternal,
4. Kesalahan perilaku atau kinerja, dan
5. Kesalahan inisialisasi dan penghentian.






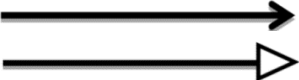
2.7 *Unified Modeling Language (UML)*

(Hasanah, F.N., & Untari, R.S, 2020) menyatakan bahwa *Unified Modeling Language (UML)* merupakan salah satu standart bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak. Intinya *UML* bisa digunakan untuk mempermudah mendokumentasikan pengembangan sebuah Aplikasi. Tipe-tipe diagram *UML* adalah sebagai berikut:

2.7.1 *Use Case Diagram*

(Hasanah, F.N., & Untari, R.S, 2020) menyatakan bahwa Use case diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem, menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Use case diagram menekankan kepada “apa” yang diperbuat oleh sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Dengan kata lain, *Use Case Diagram* menggambarkan hubungan antara sistem dan Aktor untuk menentukan sistem interaksi yang diinginkan. Simbol – symbol *Use Case* dapat dilihat pada table 2.1.

Tabel 2. 1 Use Case Diagram




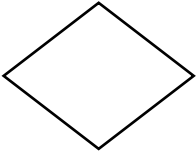

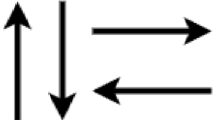
Simbol	Keterangan
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit dan aktor</p>
<p>Aktor/Actor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi.</p>
<p>Ekstensi / <i>extend</i></p> <p><<extend>></p> 	<p>Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan.</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satu lebih umum dari yang lainnya.</p>
<p>Include / <i>Use Case</i></p> <p><<include>></p> 	<p>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.</p>

2.7.2 Activity Diagram

(Hasanah, F.N., & Untari, R.S, 2020) menyatakan bahwa Activity Diagram Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi,

dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Simbol – simbol Activity Diagram dapat dilihat pada tabel 2.2.

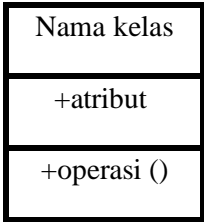
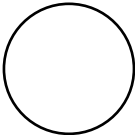


Tabel 2. 2 Activity Diagram


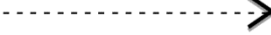

Simbol	Keterangan
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Status awal 	Bagaimana objek dibentuk atau diawali.
Status akhir 	Bagaimana objek dibentuk dan diakhiri.
Percabangan / join 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
Penggabungan / join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Garis Penghubung 	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

2.7.3 Class Diagram

(Dennis, A. 2012) menyatakan bahwa *Class Diagram* adalah model statis yang mendukung tampilan statis dari sistem yang berkembang. Ini menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam sistem lembur. Selain itu *Class diagram* juga mencerminkan kelas dan hubungan yang diperlukan untuk sebagian kasus penggunaan yang menjelaskan sistem penjualan. Simbol – simbol yang ada pada *Class Diagram* ditunjukkan pada table 2.3.

Tabel 2. 3 Class Diagram

Simbol	Keterangan
<p style="text-align: center;">Kelas</p> 	<p>Kelas pada struktur sistem</p>
<p style="text-align: center;">Antarmuka / interface</p>  <p style="text-align: center;">Nama_interface</p>	<p>Sama dengan konsep interface dalam pemograman berorientasi objek.</p>
<p style="text-align: center;">Asosisasi / <i>association</i></p> 	<p>Relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>.</p>
<p style="text-align: center;">Asosisasi berarah / <i>directed</i></p>  <p style="text-align: center;"><i>association</i></p>	<p>Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>

<p>Generalisasi</p> 	<p>Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).</p>
<p>Agregasi / <i>aggregation</i></p> 	<p>Relasi antarkelas dengan makna kebergantungan antarkelas.</p>

2.8 Penelitian Terdahulu

Berikut merupakan jurnal terkait penelitian terdahulu dapat dilihat pada table 2.4:

Tabel 2. 4 Penalitian Terdahulu

No	Nama	Judul	Terbit/Tahun	Keterangan
1	Virdyra Tasril1, Khairul, & Fachrina Wibowo	APLIKASI SISTEM INFORMASI UNTUK MENENTUKAN KUALITAS BERAS BERBASIS ANDROID PADA KELOMPOK TANI JAYA MAKMUR DESA BENYUMAS	Universitas Labuhanbatu / 2019	Penelitian ini membahas mengenai aplikasi sistem informasi berbasis android yang dapat memudahkan para kelompok tani dalam memberikan penilaian kualitas beras dan bisa digunakan dimana saja dengan menggunakan smartphone berbasis android.

2	Lilik Sumaryanti & Rosmala Widijastuti	PENGEMBANGAN SISTEM <i>E-BUSINESS</i> UNTUK UMKM PENGGILINGAN PADI	Universitas Musamus / 2020	Penelitian ini membahas mengenai manajemen transaksi pada usaha penggilingan padi, sehingga dapat meningkatkan pelayanan kepada pelanggan, dan menyediakan informasi secara real time terkait keuntungani jasai penggilingan, penjualan, pembelian, dan manajemenni data transaksi lainnyai secarai elektronik.
3	Lilik Sumaryanti & Chusnul Chotimah	SISTEM INFORMASI PERSEDIAAN PRODUKSI BERAS PADA UNIT USAHA PENGGILINGAN PADI	Universitas Musamus Merauke/ 2018	Penelitian ini membahas mengenai sistem informasi persediaan produksi.