

BAB 2

TINJAUAN PUSTAKA

2.1 Android

Android merupakan sebuah sistem operasi yang biasa digunakan untuk smartphone dan tablet. Sistem operasi dapat digambarkan sebagai “jembatan” antara perangkat (device) dan penggunanya, sehingga memungkinkan pengguna berinteraksi dengan perangkat mereka dan menjalankan aplikasi yang tersedia diperangkat. (Sari, 2016)

2.1.1 Elemen Android

1. Dalvik Virtual Machine (DVM)

Salah satu element dasar dari Android adalah Dalvik Virtual Machine (DVM). Android dijalankan di Dalvik Virtual Machine (DVM) dan bukan di Java Virtual Machine (JVM), sebenarnya ada banyak kesamaan dengan Java Virtual Machine (JVM) seperti Java ME (Java Mobile Edition), tetapi Android menggunakan mesin virtualnya sendiri yang bertujuan untuk membuat beberapa fitur-fitur berjalan lebih efisien pada perangkat mobile.

2. Android SDK (Software Development Kit)

Android SDK adalah tools API (Application Programming Interface) yang digunakan untuk memulai mengembangkan aplikasi Android menggunakan bahasa pemograman Java. Android adalah bagian dari software untuk ponsel yang mencakup sistem operasi middleware serta aplikasi utama yang diterbitkan oleh Google. Saat ini telah disediakan Android SDK (Software Development Kit) sebagai tools dan API untuk memulai pengembangan aplikasi pada platform Android menggunakan bahasa pemograman Java.

3. ADT (Android Development Tools)

Android development tools adalah plugin yang dikembangkan untuk IDE Android Studio yang memudahkan pengembang aplikasi android dalam menggunakan IDE Android Studio. Dengan menggunakan ADT untuk Android Studio akan memudahkan pembuatan aplikasi dari project Android, membuat interface aplikasi, dan menambahkan komponen lainnya.

2.2 Rancang Bangun

Rancang Bangun adalah suatu istilah umum yang digunakan untuk membuat atau mendesain suatu objek dari awal pembuatan hingga akhir konstruksi. Rancang bangun dimulai dari kata desain yang berarti perancangan, rancang, desain, bangun. Sedangkan merancang berarti menyusun, mengerjakan atau melakukan sesuatu dan perancangan berarti proses, cara, penjabaran dari design (Ariansyah, 2017)

Rancang Bangun adalah aktivitas menerjemahkan hasil analisis ke dalam bentuk paket perangkat lunak kemudian membangun sistem atau meningkatkan sistem yang sudah ada. (Elli Novia Putri, 2019)

2.3 ATM Mini (BRI-Link)

BRI Link merupakan perluasan layanan BRI dimana BRI bekerja sama dengan nasabah BRI sebagai agen yang dapat memberikan layanan real-time online banking kepada masyarakat dengan menggunakan fitur EDC mini ATM BRI dengan konsep fee-sharing. (Lathifah, 2019)

2.4 Android Studio

Android Studio merupakan IDE (Integrated Development Environment) resmi yang digunakan untuk mengembangkan aplikasi Android yang bersifat open source atau gratis. Google merilis Android Studio pada 16 Mei 2013 di konferensi I/O Google tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk pengembangan aplikasi Android (Sari, 2016)

Android Studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse dan disertai dengan ADT Plugin (Android Development Tools). Android Studio memiliki fitur:

- a. Projek berbasis pada Gradle Build
- b. Refactory dan pembenahan bug yang cepat
- c. Tools baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompatibilitas aplikasi dengan cepat.
- d. Mendukung Proguard And App-Signing untuk keamanan
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh Google Cloud Platform untuk setiap aplikasi yang dikembangkan.

2.5 MySql

MySQL merupakan sebuah implementasi dari sistem manajemen database relasional (RDBMS) yang didistribusikan secara gratis. Semua pengguna bebas menggunakan MySQL, tetapi dengan batasan perangkat lunak tidak boleh digunakan sebagai produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan dari salah satu konsep utama dari database yang sudah ada yaitu SQL (Structur Query Language). SQL adalah konsep operasi database, terutama untuk pemilihan atau seleksi data dan input, yang memungkinkan operasi data dilakukan secara otomatis dengan cara yang sederhana. (Sri Lestanti, 2016)

2.6 Laravel

Kerangka kerja adalah struktur konseptual dasar yang tujuannya adalah untuk memecahkan atau menangani masalah yang kompleks. Singkatnya framework adalah wadah atau kerangka kerja untuk sebuah aplikasi yang akan di bangun. Tujuan menggunakan kerangka kerja ini adalah mengurangi waktu yang di gunakan dalam membangun aplikasi dan membuatnya lebih mudah untuk melakukan perbaikan. Laravel merupakan framework berbasis PHP yang bersifat opensource, dan menggunakan konsep model-view-

controller. Laravel dilisensikan dibawah MIT licence dengan menggunakan github sebagai tempat menjalankan kodenya. (Naista, 2017)

2.7 API

Application Programming Interface atau API adalah integrasi dari dua bagian dari sistem aplikasi, yang terdiri dari elemen function, protocols dan tools lainnya yang digunakan pengembang untuk membangun sebuah aplikasi. Web API mendukung seperti fungsi CRUD yang bekerja melalui HTTP protocol dengan method GET, POST, PUT, dan DELETE. (Sari Noorlima Yanti, 2021)

2.8 Java

Java merupakan salah satu bahasa pemrograman yang populer dan merupakan bahasa pemrograman berorientasi objek. Bahasa pemrograman java terdiri dari beberapa bagian yang di sebut class. Salah satu kelebihan Java adalah dapat diakses di berbagai sistem operasi seperti linux, windows, Mac OS, Solaris, dan lainnya. (Wamiliana, 2015)

2.9 PHP

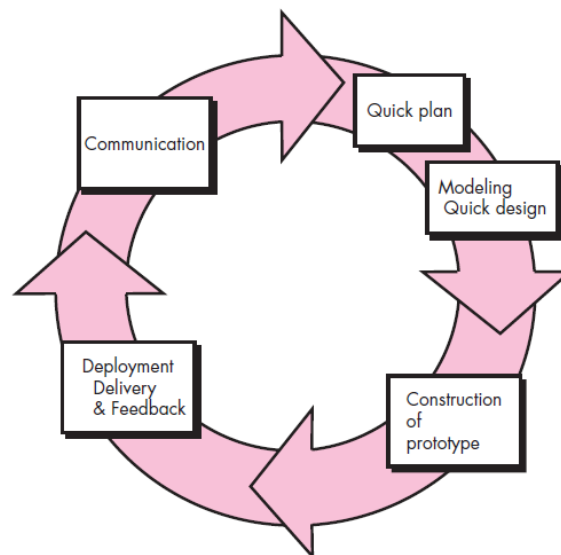
PHP (HyperText Preprocessor) digunakan sebagai bahasa script server-side dalam mengembangkan aplikasi yang disematkan didalam dokumen HTML. Penggunaan PHP memungkinkan sistem dapat dinamis sehingga pemeliharaan sistem dapat menjadi lebih mudah dan lebih efisien. (Hidayat, 2019)

2.10 Metode Pengembangan Sistem

Metodologi yang digunakan untuk membangun sistem ini adalah Model Prototype. Model ini merupakan sebuah pendekatan terhadap pengembangan perangkat lunak yang sistematis, dengan beberapa tahapan, yaitu: System Engineering, Analysis, Design, Coding, Testing dan Maintenance.

prototype adalah pengembangan yang cepat dan pengujian terhadap model kerja (prototipe) dari aplikasi baru melalui proses interaksi dan berulang-

ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis. Prototype disebut juga desain aplikasi cepat (rapid application design/RAD) karena menyederhanakan dan mempercepat desain sistem. Bagian user kesulitan mengungkapkan keinginannya untuk mendapatkan aplikasi yang sesuai dengan kebutuhannya. Kesulitan ini yang perlu diselesaikan oleh analis dengan memahami kebutuhan user dan menerjemahkannya ke dalam bentuk model (prototipe). Model ini selanjutnya diperbaiki secara terus menerus sampai sesuai dengan kebutuhan. (Pressman, 2012)



Gambar 2.1 Prototype Model Proses

Tahap pertama adalah communication dan pengumpulan data awal yaitu tahap suatu perencanaan yang dilakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, efisiensi biaya. Tahap kedua adalah quick plan yaitu analisis terhadap kebutuhan pengguna. Tahap ketiga adalah modelling quick design yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali. Tahap keempat adalah construction of prototype adalah pembuatan perangkat prototype termasuk pengujian dan penyempurnaan. Tahap kelima adalah deployment, delivery, and feedback adalah tahap penyerahan sistem ke pengguna dan umpan balik.

2.11 UML (Unified Modelling Language)

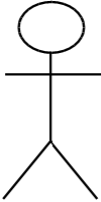
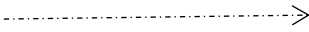
Alat bantu perancangan yang digunakan dalam penelitian ini adalah Unified Modelling Language / UML.



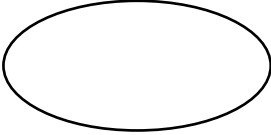
Unified Modelling Language merupakan bahasa yang digunakan untuk visualisasi, spesifikasi, pembuatan sistem perangkat lunak dan dokumentasi. UML menyediakan model yang akurat, tidak ambigu, dan lengkap. Secara khusus, UML menentukan langkah-langkah penting dalam pengembangan analisis keputusan, perancangan, serta implementasi dalam sistem perangkat lunak. (Faizal Ari Prabowo, 2017)

1. Usecase

Usecase diagram adalah model untuk membuat sistem informasi yang akan di rancang. Usecase bekerja dengan cara mendeskripsikan tipikal interaksi antara user sebuah sistem dengan sistemnya sendiri menggunakan riwayat penggunaan sistem. (Dede Wira Trise Putra, 2019)

Tabel 2.1 Simbol-Simbol Pada Usecase

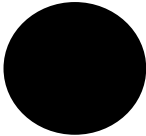
| Simbol | Keterangan |
|--|--|
| <p>Aktor</p>  | <p>mendeskripsikan himpunan peran yang dimainkan pengguna saat berinteraksi dengan usecase</p> |
| <p>Include</p>  | <p>Suatu hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (dependent) memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent)</p> |


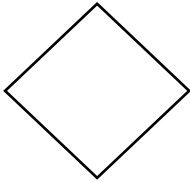

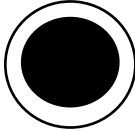
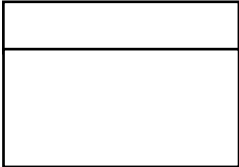
| | |
|--|--|
| Association  | menghubungkan antara objek satu dengan objek yang lainnya |
| Sistem  | Mendeskripsikan paket yang memunculkan sistem secara terbatas |
| Usecase  | Menggambarkan urutan Tindakan yang dilakukan oleh sistem yang mengarah pada hasil yang terukur bagi seorang actor. |

2. Activity Diagram

Activity diagram merupakan program yang mendeskripsikan alur kerja atau aktivitas dari sistem perangkat lunak yang ada. (Dede Wira Trise Putra, 2019)

Tabel 2.2 Simbol-Simbol pada Activity Diagram

| Simbol | Keterangan |
|---|---|
|  | Status Awal Sebuah diagram aktivitas memiliki sebuah status awal. Status awal ini menandakan dimulainya sebuah diagram aktivitas |

| | |
|---|--|
|  | <p>Aktivitas</p> <p>Aktivitas yang di lakukan oleh sistem, aktivitas biasanya dimulai dengan kata kerja</p> |
|  | <p>Decision</p> <p>Percabangan dimana ada pilihan aktivitas yang lebih dari satu. Terdapat pilihan yang akan merujuk kearah aktivitas selanjutnya.</p> |
|  | <p>Join</p> <p>Penggabungan dimana ada lebih dari satu aktivitas lalu di gabungkan menjadi satu</p> |
|  | <p>Status Akhir</p> <p>Status akhir yang di lakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. Status akhir menandakan berakhirnya sebuah diagram aktivitas.</p> |
|  | <p>Swimlane</p> <p>Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p> |

2.12 Merge Sort

Merge sort adalah metode pengurutan yang menggunakan pola divide and conquer. Caranya adalah membagi sekelompok data yang akan di urutkan

menjadi beberapa kelompok kecil terdiri dari maksimal dua nilai untuk membandingkan dan di gabungkan lagi secara keseluruhan. (Arief Hendra Saptadi)

Langkah-langkah dalam merge sort:

a. Divide

Memilah elemen-elemen dari rangkaian data menjadi dua bagian dan mengulangi pemilahan hingga satu elemen terdiri maksimal dua nilai.

b. Conquer

Mengurutkan masing-masing elemen

c. Kombinasi

Mengkombinasikan dua bagian tersebut secara rekursif untuk mendapatkan rangkaian data berurutan.

Proses rekursif berhenti jika mencapai elemen dasar. Hal ini terjadi bilamana bagian yang akan di urutkan menyisakan tepat satu elemen. Sisa pengurutan satu elemen tersebut menandakan bahwa bagian tersebut telah terurut sesuai rangkaian.

2.13 Shell Sort

Shell sort juga dikenal sebagai metode pertambahan menurun (diminishing increment). Metode ini dikembangkan oleh Donald L. Shell pada tahun 1959, oleh karena itu sering disebut sebagai metode Shell sort. Dengan menggunakan metode ini, pengurutan dilakukan dengan membandingkannya dengan data lain pada jarak tertentu dan menggantinya jika perlu. (Heru Satria Tambunan, 2018)

2.14 Penelitian Sebelumnya

Tabel 2.3 Literatur Review

| Judul | Penulis | Pembahasan |
|-------------------------------------|--------------------------------------|---|
| Optimasi Algoritma Shell Sort Dalam | Heru Satria Tambunan, Sumarno, Indra | Latar Belakang Disetiap algoritma dalam perancangan |

| | | |
|--|----------------------------|---|
| <p>Pengurutan Data Huruf dan Angka</p> | <p>Gunawan, Eka Irawan</p> | <p>perangkat lunak pasti membutuhkan sebuah data, baik data analog maupun data digital, yang mana data tersebut banyak berupa numeric bahkan huruf yang akan dikelola sehingga menjadi hasil yang diinginkan. Dari data yang digunakan berupa data acak, yang mana data tersebut sulit untuk diurutkan karena adanya data berupa huruf dan angka. Pengurutan data atau yang disebut dengan sorting proses menyusun kembali data berupa numerik ataupun karakter yang sebelumnya telah disusun dengan suatu pola tertentu, sehingga tersusun secara teratur menurut aturan tertentu. Baik itu secara ascending</p> |
|--|----------------------------|---|

| | | |
|--|--|--|
| | | <p>maupun dengan descending Beberapa metode pengurutan data atau sorting yaitu Pengurutan berdasarkan perbandingan (comparison-based sorting) yang memiliki jenis pengurutan yakni Bubble sort, exchange sort.</p> <p>Untuk pengurutan berdasarkan prioritas (priority queue sorting method) yakni Selection sort, heap sort (menggunakan tree), untuk pengurutan berdasarkan penyisipan dan penjagaan terurut (insert and keep sorted method) yakni Insertion sort, tree sort, sedangkan pengurutan berdasarkan pembagian dan</p> |
|--|--|--|

| | | |
|--|--|---|
| | | <p>penguasaan (divide and conquer method) yakni Quick sort, merge sort, dan yang terakhir adalah pengurutan berkurang menurun (diminishing increment sort method) yaitu Shell sort (pengembangan insertion) yang mana masing-masing pengurutan tersebut memiliki kelebihan. Algoritma Shell Sort merupakan juga dengan metode penambahan menurun (diminishing increment).</p> |
| <p>Analisis Algoritma Insertion Sort, Merge Sort dan Implementasinya Dalam Bahasa Pemograman C++</p> | <p>Arief Hendra Saptadi, Desi Windi Sari</p> | <p>Untuk mengetahui seberapa baik performa dalam mengurutkan data, maka dalam tahap terakhir, kedua algoritma tersebut mengurutkan sejumlah angka acak dengan rentang jumlah yang sudah</p> |

| | | |
|---|-------------------------------|--|
| | | <p>ditentukan dan hasilnya kemudian dibandingkan. Dari eksperimen yang sudah dilakukan, algoritma merge sort telah memperlihatkan performa yang lebih baik, khususnya untuk jumlah data yang banyak (> 10000). Adapun algoritma insertion sort memiliki keuntungan dalam hal kompleksitas algoritma yang lebih rendah terutama dalam kondisi best case dan karena tidak menggunakan rutin rekursi dalam proses pengurutan, maka tidak membutuhkan ruang penyimpanan atau memori sebanyak algoritma Merge Sort.</p> |
| <p>Perbandingan Kecepatan Gabungan Algoritma Utama Quick Sort dan Merge</p> | <p>Muhammad Ezar Al Rivan</p> | <p>Algoritma Quick Sort dan Merge Sort merupakan algoritma yang memiliki</p> |

| | | |
|---|-----------------------------|--|
| <p>Sort Dengan Algoritma Tambahan Insertion Sort, Bubble Sort, dan Selection Sort</p> | | <p>keunggulan dalam melakukan pengurutan dengan jumlah data yang besar. Algoritma Insertion Sort, Bubble Sort dan Selection Sort memiliki keunggulan dalam mengurutkan data dengan jumlah kecil. Proses pengurutan sub-sub bagian kecil ini akan diganti dengan menggunakan algoritma Insertion Sort, Bubble Sort dan Selection Sort. Hasil penelitian menunjukkan bahwa Merge Sort lebih cepat dibandingkan Insertion Sort untuk data dalam jumlah besar.</p> |
| <p>Kompleksitas Algoritma Quick Sort Guna Menemukan Efisiensi Waktu dan Memori</p> | <p>Wahju Tjahjo Saputro</p> | <p>Dari sekian algoritma, salah satu metode algoritma yang sering digunakan yaitu Algoritma Quick Sort. Dalam teknik tersebut</p> |

| | | |
|--|--|---|
| | | <p>dapat ditentukan pivot dari kiri, kanan atau tengah. Maka dapat dikatakan algoritma adalah langkah yang diambil dalam menyelesaikan pekerjaan.</p> <p>Ada dua macam kompleksitas algoritma yaitu: kompleksitas waktu dan kompleksitas ruang. Kompleksitas waktu disimbolkan dengan $T(n)$ dan kompleksitas ruang $S(n)$. Kompleksitas waktu, $T(n)$, diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari ukuran masukan n.</p> <p>Sedangkan kompleksitas ruang, $S(n)$, diukur dari memori yang digunakan oleh</p> |
|--|--|---|

| | | |
|--|------|---|
| | | <p>struktur data yang terdapat di dalam algoritma sebagai fungsi dari ukuran masukan n. Dengan menggunakan besaran kompleksitas waktu atau ruang algoritma, dapat menentukan laju peningkatan waktu, dalam hal ini ruang, yang diperlukan algoritma dengan meningkatnya ukuran masukan n.</p> |
| <p>Komparasi Algoritma Quicksort dan Bucketsort Pada Pengurutan Data Integer</p> | Audy | <p>Pengolahan data erat kaitannya dengan pencarian data, dimana dalam pencarian tersebut terdapat proses memilah-milah data sesuai kebutuhan. Pencarian data yang efektif dan efisien tidak dapat dilepaskan dari faktor keterurutan data. Data yang sudah terurut akan mempermudah dan mempercepat</p> |

| | | |
|--|--|--|
| | | <p>pencarian data. Oleh karena itu, dibutuhkan suatu algoritma yang dapat mengurutkan data secara benar, efektif, dan efisien.</p> <p>penulis ingin melakukan komparasi antara dua algoritma pengurutan, yaitu Quicksort dan Bucket sort, dalam mengurutkan data bertipe integer dan membandingkan performa yang dihasilkan oleh kedua algoritma tersebut, baik dari sisi waktu maupun penggunaan memori</p> |
|--|--|--|