

BAB II TINJAUAN PUSTAKA

2.1 Algoritma *Brute force*

Salah satu Algoritma *String Matching* adalah algoritma *Brute force*. Cara kerja algoritma ini adalah dengan membandingkan satu-persatu antara karakter di teks dan di pola dari kiri ke kanan. Misalnya i menyatakan *indeks* pada teks (antara 0 sampai $n-1$) dan j menyatakan *indeks* dari pola (antara 0 sampai $m-1$). *Window* diletakkan di posisi paling kiri dari teks, lalu lakukan perbandingan pertama, yaitu antara $T[0]$ dan $P[0]$. Jika terjadi kecocokan maka masing-masing *indeks* akan dinaikkan satu, jadi jika misalnya $T[0] = P[0]$, maka i dan j dinaikkan satu sehingga selanjutnya adalah membandingkan $T[1]$ dan $P[1]$. Jika terjadi ketidakcocokan maka *indeks* j akan dikembalikan ke awal pola yaitu $j = 0$ dan *window* akan digeser satu ke kanan sehingga *indeks* i sama dengan satu, $i = 1$. Sehingga perbandingan selanjutnya adalah antara $T[1]$ dan $P[0]$. Hal ini adalah ciri algoritma *brute force*, yaitu jika terjadi ketidakcocokan maka *window*-nya pasti akan digeser ke kanan sebanyak satu.

Perbandingan ini akan dilakukan sampai batas kanan dari *window* melebihi dari batas kanan teks. Jika sampai pada akhir dari pola ($j = m-1$) maka artinya terdapat pola yang dicari pada teks, yang dimulai dari $T[i - m]$. sedangkan jika akhir dari teks dicapai sebelum akhir dari pola dicapai maka pola yang dicari tidak ada pada teks (Darmawan Utomo & Eric Wijaya Harjo, 2008).

2.2 *Android*

Arfida, Amnah dan Wibowo (2018 : 52) menguraikan bahwa *Android* merupakan sebuah sistem operasi yang berbasis Linux untuk perangkat *portable* seperti *smartphone* dan computer tablet. *Android* menyediakan *platform* terbuka bagi *programmer* untuk mengembangkan Aplikasi sendiri pada berbagai perangkat dengan sistem operasi *Android*. *Android* merupakan sistem operasi untuk telepon seluler berbasis linux sebagai kernelnya. *Android* menyediakan *platform* terbuka (*Open Source*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri.

2.3 Sistem Operasi

Ariyus dan Pangera (2010 : 57) menguraikan bahwa Sistem Operasi merupakan sebuah penghubung antara pengguna mesin dengan perangkat keras yang dimiliki mesin tersebut. Sebelum ada sistem operasi, orang hanya menggunakan komputer dengan menggunakan sinyal analog dan digital. Seiring dengan berkembangnya pengetahuan dan teknologi, pada saat ini terdapat berbagai sistem operasi dengan keunggulan masing-masing. Sistem operasi bertindak sebagai antarmuka antara program aplikasi dengan perangkat keras komputer, *level* dari pengguna setiap lapisan juga berbeda-beda. Program aplikasi hanya digunakan oleh pemakai terakhir (*End User*), sedangkan sistem operasi dan perangkat keras merupakan tugas pemrogram dan pendesain sistem operasi tersebut.

2.4 Aplikasi

Rahman dan Santoso (2015 : 79) menguraikan bahwa Aplikasi merupakan sekelompok atribut yang terdiri dari beberapa *form, report* yang disusun sedemikian rupa sehingga dapat mengakses data. Aplikasi merupakan program yang berisi perintah untuk melakukan olah data, secara umum aplikasi adalah suatu proses dari cara manual yang dipindahkan ke dalam komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Karakteristik perangkat *mobile* adalah sebagai berikut :

1. Ukuran yang kecil : Perangkat *mobile* memiliki ukuran yang kecil. Para pengguna menginginkan perangkat yang kecil untuk kenyamanan
2. Memory yang terbatas : Perangkat *mobile* juga memiliki memory yang kecil, yaitu *Primary* (RAM) dan *Secondary* (Disk)
3. Mengonsumsi Daya yang Rendah : Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop

2.5 Perangkat Lunak yang Digunakan

2.5.1 Android Studio

Juansyah (2015 : 17) menguraikan bahwa *Android studio* adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi *Android* dan bersifat open source atau gratis. *Android Studio* menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi *Android*. Yudhanto dan Wijayanto (2017 : 17) *Android studio* ini bersifat free dibawah Apache Lincense 2.0. *Android Studio* awalnya bermula dengan versi 0.1 pada mei 2013, kemudian dibuat versi beta 0.8 yang dirilis 2014. Berbasiskan JetBrainns' IntelliJ IDEA, Studio di desain khusus untuk *Android Development*. *Android studio* memiliki fitur :

1. Projek berbasis pada Gradle Build
2. *Refactory* dan pembenahan bug yang cepat
3. *Tools* baru yang bernama "Lint" dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
4. Mendukung Proguard And *App-signing* untuk keamanan.
5. Memiliki GUI aplikasi *Android* lebih mudahf. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

2.5.2 Java

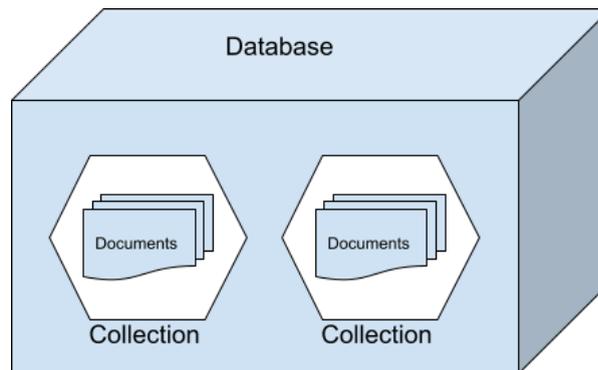
Harison, Busran (2016 : 2) menguraikan java adalah bahasa pemrograman serbaguna. dapat digunakan membuat suatu program, mendukung sumber daya internet yang saat ini populer, yaitu *World Wide Web* atau sering disebut Web saja. Mendukung aplikasi klien / *server*, baik dalam jaringan lokal (LAN) maupun WAN.

2.5.3 Android Software Development Kit (SDK)

Safaat (2011:5) menguraikan bahwa *Software Development Kit* (SDK), adalah tools API (*Aplication Programming Interface*) yang diperlukan untuk memulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman java. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleweredan aplikasi kunci yang di *release* oleh Google.

2.6 *MongoDB*

MongoDB adalah salah satu jenis database *NoSQL* yang cukup populer digunakan dalam pengembangan website. Berbeda dengan database jenis *SQL* yang menyimpan data menggunakan relasi tabel, *MongoDB* menggunakan dokumen dengan format JSON. Hal inilah yang dianggap membuat pengelolaan data menggunakan *MongoDB* lebih baik. Alhasil, banyak perusahaan besar seperti *Adobe*, *Google* dan *ebay* yang menggunakannya. Sistem data base ini menggunakan beberapa komponen penting, yaitu:



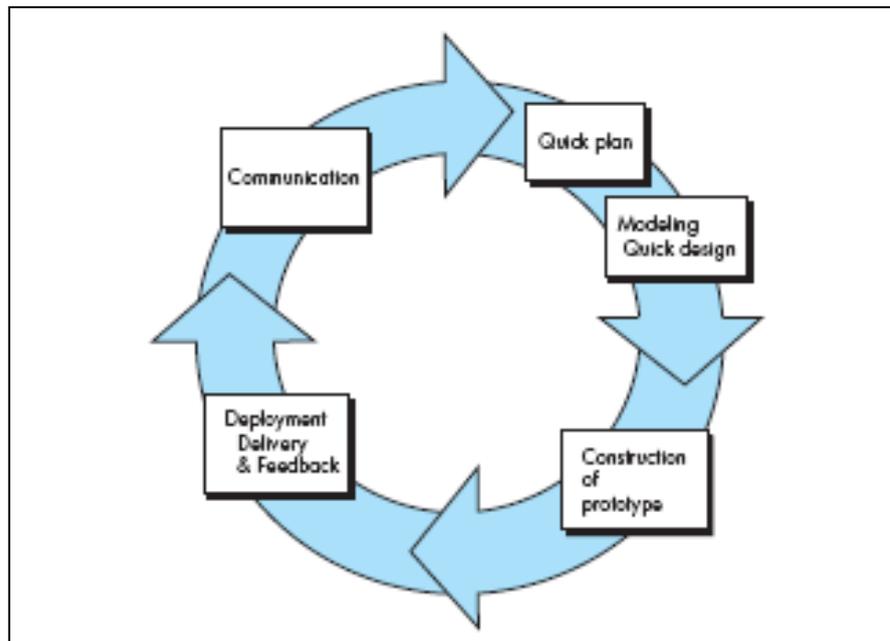
- Database** merupakan wadah dengan struktur penyimpanan yang disebut *collection*.
- Collection** merupakan tempat kumpulan informasi data yang berbentuk dokumen. Collection dipadankan seperti tabel-tabel yang berisi data pada database SQL.
- Document** merupakan satuan unit terkecil dalam *MongoDB*.

2.7 **Basis Data / Database**

Fathansyah (2012:12) menguraikan bahwa Sistem basis data merupakan sistem yang terdiri dari atas kumpulan tabel yang salingberhubungan (dalam sebuah basis data disebut sistem komputer) dan sekumpulan program (yang biasa disebut DBMS /Data Base Management System) yang memungkinkan beberapa pemakaidan/atau program lain untuk mengakses danmemanipulasi tabel-tabel data tersebut.

2.8 Metode *Prototype*

Sukamto & Shalahuddin (2015:33) menguraikan bahwa Model *prototype* memberikan kesempatan untuk pengembang program dan objek penelitian untuk saling berinteraksi selama proses perancangan sistem. Metode pengembang perangkat lunak yang digunakan dalam aplikasi ini adalah metode *prototype*. Model *prototype* digunakan untuk merancang sistem informasi. Gambar 2.1 berikut menjelaskan tahap – tahapan *Prototype* :



Gambar 2. 1 Tahapan-Tahapan *Prototype*

Tahapan dari model *Prototype* adalah sebagai berikut :

1. *Communication* (Komunikasi) : pengumpulan data awal, yaitu komunikasi dengan klien dan user untuk menentukan kebutuhan.
2. *Quick Plan* (Perencanaan Cepat) : pembuaan perencanaan analisis terhadap kebutuhan pengguna.
3. *Modeling Quick Design* (Pemodelan Perancangan Cepat) : membuat rancangan desain program.
4. *Construction of prototype* (Pembentukan Prototype) : pembuatan aplikasi berdasarkan dari pemodelan desain yang telah dibuat.
5. *Deployment Deelivery and Feedback* (Penyerahan Sistem dan Umpan Balik) : memproduksi perangkat ssecara benar sehingga dapat digunakan oleh pengguna.

1.9 Pengujian *Black-Box Testing*

Roger S. Pressman (2010) *Black Box Testing* atau Pengujian Kotak Hitam atau juga disebut *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black-Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. *Black-Box Testing* bukan merupakan alternatif dari pengujian *White Box Testing*. Sebaliknya, *Black-Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White Box Testing*.

1.10 Sistem Pemodelan

2.10.1 *Unified Modeling Language*

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat *Software* berorientasi objek. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma object oriented. UML adalah salah satu tool / model untuk merancang pengembangan *Software* yang berbasis object oriented. UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *Software* (Adi Nugroho, 2004). Terdapat beberapa bagian dalam perancangan UML yaitu sebagai berikut:

2.10.2 Use Case Diagram

Use case diagram adalah pemodelan untuk menggambarkan perilaku sistem yang akan dibuat pada sebuah aplikasi. Komponen- komponen *use case* dapat dilihat pada tabel 2.1:

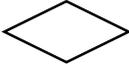
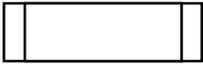
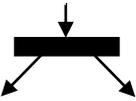
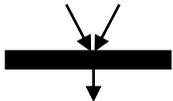
2. 1 Simbol *Use Case Diagram*

	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2.		<i>Include</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		Sistem	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
5.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

2.10.3 Activity Diagram

Diagram aktivitas merupakan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem yang di gambarkan dalam bentuk diagram. Diagram aktivitas adalah penggambaran sebuah aktivitas sistem bukan apa yang dikerjakan actor, jadi aktivitas dilakukan oleh sistem. simbol-simbol dari *activity* diagram dapat dilihat pada tabel 2.2:

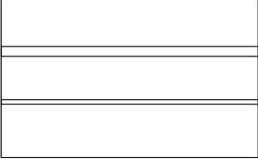
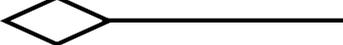
Tabel 2. 2 Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal dari diagram aktivitas untuk mengawali proses aktivitas sistem.
Aktivitas 	Aktivitas pada sistem biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan ada jika pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dilakukan jika ada lebih dari satu aktivitas dan digabungkan menjadi satu.
Status akhir 	Status akhir dari sistem yang merupakan akhir dari suatu sistem aktivitas.
<i>Database</i> 	<i>Symbol</i> ini untuk menunjukkan penyimpanan pada database sistem.
<i>Fork</i> , 	Menunjukkan aktivitas yang dilakukan secara parallel.
<i>Join</i> 	Menunjukkan aktivitas yang digabungkan.

2.10.4 Class Diagram

Class Diagram merupakan penggambaran struktur sistem dari segi pendefinisian class-class yang akan dibuat dalam membangun sistem. Penjelasan simbol class diagram pada tabel 2.3.

Tabel 2. 3 *Class Diagram*

Simbol	Keterangan
<p><i>Class</i></p> 	Menambahkan kelas baru pada Diagram
<p><i>Association</i></p> 	Menggambar relasi asosiasi
<p><i>Generalization</i></p> 	Menggambar relasi generalisasi
<p><i>Realize</i></p> 	Menggambar relasi realisasi
<p><i>Association class</i></p> 	Menghubungkan kelas asosiasi (association class) pada suatu relasi asosiasi
<p><i>Return message</i></p> 	Menggambarkan pengembalian dari pemanggilan prosedur
<p><i>Aggregation</i></p> 	Menggambarkan relasi agregasi

2.11 Penelitian Terkait

Penelitian terdahulu dilakukan dengan mengumpulkan literatur yang bersumber dari jurnal atau hasil penelitian orang lain yang berkaitan dengan objek penelitian ini. Berikut Penelitian terdahulu dapat dilihat pada tabel 2.4.

Tabel 2. 4 Penelitian Terkait

No	Nama	Judul	Sumber	Uraian
1.	Abdul Azis M Saleh	Pembangunan Aplikasi Penghubung Donatur Dan Panti Asuhan Berbasis Lokasi	e-journal.uajy (2017)	<p>Penelitian ini bertujuan untuk mempermudah donatur untuk menyalurkan bantuan atau donasi kepada panti asuhan sesuai dengan kebutuhan panti asuhan, dengan demikian donatur tidak akan bingung dalam menyalurkan bantuannya</p> <p>Keterbaruan :</p> <p>Keterbaruan pada penelitian ini yaitu menerafkan algoritma <i>brute force</i> untuk mempermudah donatur untuk menyalurkan bantuan atau donasi sesuai dengan kebutuhan.</p>
2.	Muhamad Ikhsan, NovriA dhiatma, Dedi Kurniady	perancangan sistem informasi donasi berbasis <i>Android</i> (studi kasus yayasan as sunnah jambi)	ejournal.nusa mandir (2018)	Penelitian ini yaitu membangun sistem informasi donasi berbasis hybrid <i>mobile</i> dengan menggunakan web service yang dapat memberikan layanan

				<p>komunikasi dan promosi kepada donatur</p> <p>Keterbaruan :</p> <p>Di penelitian ini memiliki pembaruan yaitu tidak berbasis web namun berbasis <i>Android</i> dan menerafkan algoritma <i>brute force</i> untuk mempermudah donatur untuk menyalurkan bantuan atau donasi sesuai dengan kebutuhan.</p>
3.	Sopings	Sistem Informasi Donasi Berbasis Hybrid <i>Mobile</i> Menggunakan Web Service pada Yayasan Solo Peduli	DutaCom, (2016)	<p>Penelitian ini yaitu membangun sistem informasi donasi berbasis hybrid <i>mobile</i> dengan menggunakan web service yang dapat memberikan layanan komunikasi dan promosi kepada donatur</p> <p>Keterbaruan :</p> <p>Di penelitian ini memiliki pembaruan yaitu tidak berbasis web namun berbasis <i>Android</i> dan menerafkan algoritma <i>brute force</i> untuk mempermudah donatur untuk menyalurkan</p>

				bantuan atau donasi sesuai dengan kebutuhan.
4.	Dewi Rizki Anggraini	<i>Rancang Bangun Aplikasi Donasi Berbasis Website pada BSMI Cabang Surabaya</i>	Jurnal Skripsi universitas dinamika (2018)	<p>Penelitian ini bertujuan untuk membantu lembaga Bulan Sabit Merah Indonesia cabang Surabaya dalam mengatasi kendala dan permasalahan yang terjadi pada proses donasi.</p> <p>Keterbaruan : Di penelitian ini memiliki pembaruan yaitu tidak berbasis web namun berbasis <i>Android</i> dan menerapkan algoritma <i>brute force</i> untuk mempermudah donatur untuk menyalurkan bantuan atau donasi sesuai dengan kebutuhan.</p>