

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini didasarkan pada latar belakang tentang sulitnya mencari informasi tentang aplikasi kamus kata serapan yang dengan model pencarian yang mudah dan tersusun secara berurutan (*sequential*) maka digunakan tinjauan studi yang dapat mendukung penelitian, berikut ini merupakan tinjauan studi yang diambil yaitu:

Tabel 2.1 Tinjauan Pustaka

No	Penulis	Judul	Solusi	Hasil
1	(Aziz &	Aplikasi Kamus Bahasa Betawi Berbasis Androin Menggunakan Metode Sequential Search	Untuk mengatasi masalah tersebut dibuat aplikasi kamus betawi online	Aplikasi Kamus Bahasa Betawi Berbasis Mobile
2	(Gunawan, 2016)	Aplikasi Kamus Istilah Ekonomi (Inggris – Indonesia) Menggunakan Metode Sequential Searching	Merancang aplikasi kamus online untuk mempermudah pencarian dan mengurangi biaya pembelian kamus dalam bentuk buku	Aplikasi Kamus istilah Ekonomi (Inggris – Indonesia)
3	Hizbulah Salim, Fitri Marisa, Indra Dharma Wijaya	Aplikasi Kamus Istilah Neurology Berbasis Mobile Menggunakan Metode Sequential Search	Merancang Aplikasi kamus Neurology yang lebih efisien agar mudah di gunakan	Aplikasi Kamus Istilah Neurology Berbasis Mobile
4	(Sembiring,	Perancangan Aplikasi Kamus Bahasa	Merancang sebuah	Aplikasi Kamus

		Indonesia – Karo Online Berbasis Web dengan Metode Sequential Search	aplikasi yang dapat mempercepat pencarian data	Bahasa Indonesia – Karo Online Berbasis Web
--	--	--	--	---

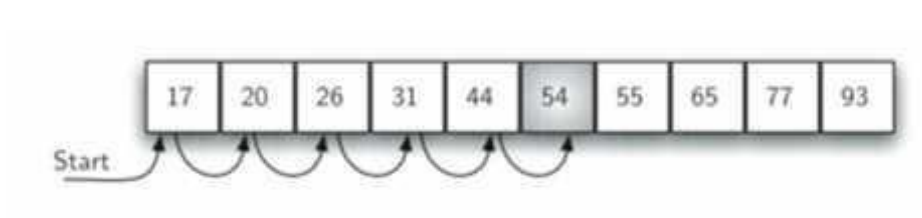
2.2 Rancang Bangun

Perancangan adalah proses yang mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaan.

Bangun sistem adalah kegiatan menciptakan sistem baru mau pun pengganti atau memperbaiki sistem telah ada secara keseluruhan (Pressman, 2002). Rancang bangun berarti mengatur segala sesuatu (sebelum bertindak, mengerjakan, atau melakukan sesuatu), merencanakan (Purwanto, 2008).

2.3 Metode *Sequential Search*

Pencarian berurutan sering disebut pencarian linear merupakan metode pencarian yang paling sederhana. Pencarian berurutan menggunakan prinsip sebagai berikut : data yang ada dibandingkan satu persatu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan. Konsepnya yaitu dengan melakukan perbandingan data satu persatu secara berurutan sampai data tersebut ditemukan ataupun tidak ditemukan.



Gambar 2.1 Konsep *Sequential Search*

Pada dasarnya, pencarian ini hanya melakukan pengulangan dari 1 sampai dengan jumlah data. Pada setiap pengulangan, dibandingkan data ke-i

dengan yang dicari. Apabila sama, berarti data telah ditemukan. Sebaliknya apabila sampai akhir pengulangan tidak ada data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk N elemen data harus dilakukan pencarian sebanyak N kali pula. (Sembiring, 2013). Berikut Adalah Penjelasan dari Algoritma Sequential Search

```

i ← 0
Ketemu ← False
Jika (Data [i] = Key) Maka
    Ketemu ← True
Jika Tidak
    i ← i+1
Jika (Ketemu) Maka
    i adalah indeks dari data yang di cari
Jika tidak
    Data tidak ditemukan

```

Gambar 2.2 Algoritma Sequential Search

Pada algoritma sequential search ini tahapan proses pencariannya adalah sebagai berikut :

1. Kata yang terdapat didalam sistem atau database akan dimasukan kedalam bentuk array satu dimensi yang dimulai dari index ke 0 hingga index sesuai jumlah data kata yang ada pada sistem.
2. Setiap kata yang akan dicari kan dibandingkan satu persatu dengan array data kata yang dimulai dari index ke 0.
3. Jika kata yang dicari tidak sama dengan kata pada index ke 0 maka perbandingan akan dilanjutkan ke index berikutnya atau index + 1, proses berlanjut hingga index terakhir jika data tidak ditemukan.
4. Jika pada index berikutnya ditemukan kata yang sesuai dengan kata yang dicari maka proses pencarian selesai .

5. Jika data yang dicari tidak ditemukan hingga index terakhir pada data array, maka data tidak ditemukan.

2.4 Kamus

Menurut (Fadlullah, 2012) kamus adalah buku acuan yang memuat kata dan ungkapan yang biasanya disusun menurut abjad berikut keterangan tentang maknanya, pemakaiannya dan terjemahannya. Kamus dapat juga digunakan sebagai buku rujukan yang menerangkan makna kata-kata yang berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mungkin mempunyai pedoman sebutan, asal-usul (*etimologi*) sesuatu perkataan dan juga contoh penggunaan bagi sesuatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus terdapat banyak kamus yang populer di Indonesia, seperti : kamus bahasa Inggris, bahasa Jerman, bahasa mandarin, bahasa Jepang dan lain sebagainya.

2.5 Kata Serapan Asing

Kata Serapan Asing adalah kata yang berasal dari bahasa lain (bahasa daerah/bahasa luar negeri) yang kemudian ejaan, ucapan, dan tulisannya disesuaikan dengan penuturan masyarakat Indonesia untuk memperkaya kosa kata. Setiap masyarakat bahasa memiliki tentang cara yang digunakan untuk mengungkapkan gagasan dan perasaan atau untuk menyebutkan atau mengacu ke benda-benda di sekitarnya. Hingga pada suatu titik waktu, kata-kata yang dihasilkan melalui kesepakatan masyarakat itu sendiri umumnya mencukupi keperluan itu, namun manakala terjadi hubungan dengan masyarakat bahasa lain, sangat mungkin muncul gagasan, konsep, atau barang baru yang datang dari luar budaya masyarakat itu. Dengan sendirinya juga diperlukan kata baru. Salah satu cara memenuhi keperluan

itu yang sering dianggap lebih mudah adalah mengambil kata yang digunakan oleh masyarakat luar yang menjadi asal hal ihwal baru itu

2.6 Android

Android merupakan sistem operasi untuk perangkat mobile yang berbasis Linux dan bersifat terbuka atau opensource dengan lisensi GNU yang dimiliki Google (Wahana, 2013). Awalnya, *Google Inc.* membeli *Android Inc.* pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance* konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi.

2.7 Perangkat Lunak

2.7.1 MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB yang pada saat itu bernama TcX DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya TcX membuat MySQL dengan tujuan mengembangkan aplikasi web untuk klien. TcX merupakan perusahaan pengembang *software* dan konsultan *database*. Saat ini MySQL sudah diakuisisi oleh Oracle Corp. MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang mana *database* sebagai sumber dan pengelolaan datanya. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses database sehingga mudah untuk digunakan. MySQL juga bersifat *open source* dan *free* pada berbagai *platform* kecuali pada *windows* yang bersifat *shareware*. MySQL didistribusikan dengan lisensi open source GPL (*General Public License*) mulai versi 3.23, pada bulan Juni 2000. Software MySQL bisa diunduh di <http://mysql.org> atau <http://www.mysql.com>. (Arief, 2011)

2.7.2 Java

Menurut (Shelly, B. Gary, 2012), Java adalah bahasa programming komputer tingkat tinggi. Bahasa programming tingkat tinggi seperti java membuat programmers dapat menuliskan instruksi menggunakan English- like command dan kata-kata daripada menggunakan cryptic numeric codes atau memory addresses.

Java didesain pada awal tahun 1990-an oleh tim dari Sun Microsystems yang dipimpin oleh James Gosling. Desainer java memulai dari syntax dasar seperti C, C++, dan Smalltalk. Pada awalnya Java didesain untuk penggunaan pada devices seperti telepon selular, akan tetapi beberapa tahun kemudian, Sun Microsystems menggunakan Java untuk animasi dan interaktifitas di dalam Web.

2.7.3 Android Studio

Android Studio adalah sebuah Integrated Development Environment (IDE) untuk platform Android. Android Studio tersedia bagi pengembang untuk mencoba secara gratis. Android Studio berada di awal tahap preview access mulai dari versi 0.1 pada Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada bulan Juni 2014. Berdasarkan software JetBrains 'IntelliJ IDEA, Android Studio dirancang khusus untuk pengembangan Android. ([http://en.wikipedia.org/wiki/Android Studio](http://en.wikipedia.org/wiki/Android_Studio)).

2.7.4 Android SDK (Software Development Kit)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu

dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan.

2.7.5 ADT (Android Development Tools)

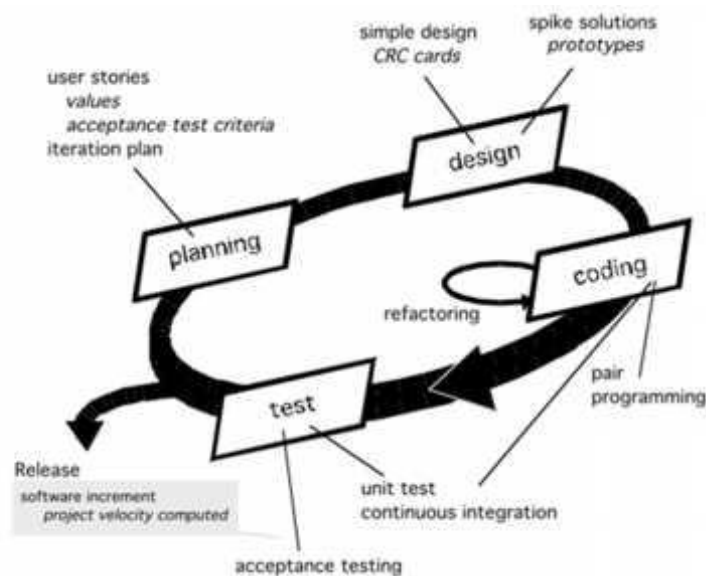
Android Development Tools adalah *plugin* yang di desain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya. Berikut Adalah Tabel Versi Android :

Tabel 2.2 Tabel Versi Android

Versi	Nama	Rilis	Catatan
1.0	-	23-Sep-08	Android pertama, hanya untuk <i>smartphone</i>
1.1	-	09-Feb-09	
1.5	Cupcake	30-Apr-09	Mulai pakai kode nama
1.6	Donut	15-Sep-09	
2.0-2.1	Eclair	26-Okt-09 (2.0)	
		12-Jan-10 (2.1)	
2.2	Frozen Yogurt	20-Mei-10	
2.3	Gingerbread	06-Des-10	Digunakan <i>smartphone jenis lama</i>
3.0-3.2	Honeycomb	22-Feb-11 (3.0)	Hanya untuk tablet
		10-Mei-11 (3.1)	
		15-Jul-11 (3.2)	
4.0	ICS	19-Okt-11	<i>Smartphone & Tablet</i>
4.1-4.3	Jelly Bean	09-Jul-12 (4.1)	Update fitur-fitur ICS
		13-Nov-12 (4.2)	
		24-Jul-13 (4.3)	
4.4	Kit Kat	Okt-13	

2.8 Metode Pengembangan Sistem

Menurut (Pressman, 2002) extreme programming adalah extreme programming merupakan salah satu dari agile software development yang banyak paling banyak digunakan. terdapat 5 (lima) nilai yang menjadi fondasi dalam melakukan kegiatan dalam extreme programming, adalah communication, simplicity, feedback, courage, respects. (Pressman, 2002) menyatakan bahwa dalam model extreme programming memiliki 4 (empat) kerangka kerja yang dilakukan yaitu planning, design, coding dan testing. (Pressman, 2002) menggambarkan metode extreme programming di tampilan pada gambar di bawah ini. Berikut adalah gambar model extreme programming:



Gambar 2.3 Model Extreme Programming

Berikut tahapan dalam model Extreme Programming:

1. Planning, kegiatan planning dilakukan dengan mengumpulkan user stories, yang menjelaskan mengenai output, fitur dan fungsionalitas dari software yang dibuat.

2. Design, design pada XP mengikuti prinsip KIS (Keep it Simple). Design yang dibuat lebih sederhana karena lebih banyak disukai dibandingkan design yang complex.
3. Coding, XP diawali dengan melakukan serangkaian tes, tes yang dilakukan harus berfokus pada implementasi untuk melewati tes.
4. Testing, testing dilakukan melalui pengujian kode pada unit testing. XP juga merekomendasikan untuk dilakukan customer test, yaitu pengujian yang dilakukan oleh customer yang berfokus kepada fitur dan fungsi dari sistem secara keseluruhan. Customer test ini berasal dari user stories yang telah diimplementasikan.

2.9 Unified Modeling Language (UML)





Unified Modeling Language (UML) adalah sebuah Teknik pengembangan sistem yang menggunakan Bahasa garfish sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada system (Mulyani, 2016). Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. Menurut (Mulyani, 2017) tipe-tipe diagram UML adalah sebagai berikut :

2.9.1 Use Case Diagram

Use case diagram adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mepresentasikan sebuah interaksi antara actor dan sistem.

Dalam *use case* diagram terdapat istilah seperti aktor, *use case* dan *case relationship*

Tabel 2.3 Komponen *Use Case Diagram*


Simbol	Nama elemen	Keterangan
	Aktor	Merupakan kesatuan eksternal yang berinteraksi dengan sistem
	Use Case	Rangkaian / uraian sekelompok yang saling terkait dan membentuk sistem
	Asosiasi	Penghubung antar elemen (aktor/use case) didalam sistem.
	Generalisasi	Sebuah elemen yang dihasilkan dari pewarisan elemen lain.
<<extend>>	Extend	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tamahan itu.
<<include>>	Include	<i>Use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

2.9.2 Activity Diagram

Activity Diagram yang disediakan oleh UML melengkapi *use case* yang telah dibuat sebelumnya memberikan representasi grafis dari aliran-aliran interaksi di dalam suatu skenario yang sifatnya spesifik. Mirip dengan diagram alir, suatu diagram aktifitas menggunakan sebuah kotak yang berisi lengkung untuk menggambarkan fungsi tertentu yang ada dalam suatu sistem yang akan dikembangkan, sementara itu tanda panah menggambarkan aliran didalam sistem dan seterusnya.

Tabel 2.4 Komponen *Activity Diagram*

Simbol	Keterangan
	<i>Activity</i> : Memperlihatkan

	bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial Node</i> : Bagaimana objek dibentuk atau diawali
	<i>Activity Final Node</i> : Bagaimana objek dibentuk dan diakhiri.
	<i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
	<i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	<i>Join</i> : Digunakan untuk menunjukkan kegiatan yang digabungkan.
	<i>Fork</i> : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel





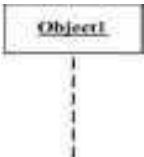
2.9.3 Sequence Diagram

Sequence diagram di UML terutama digunakan untuk memodelkan interaksi antara aktor dan objek dalam sistem dan interaksi antara obyek itu sendiri. UML memiliki sintaks yang kaya untuk sequence diagram, yang memungkinkan berbagai jenis interaksi yang dimodelkan. Sesuai namanya, sequence diagram menunjukkan urutan interaksi yang terjadi antara use case. Sequence diagram memiliki dua buah karakteristik yaitu :

1. Setiap objek memiliki lifeline yang digambarkan dengan garis putus-putus vertikal dan garis ini menunjukkan daur hidup dari sebuah objek.

2. Terdapat fokus kontrol yang digambarkan dengan sebuah persegi panjang yang tipis dan tinggi. Fokus kontrol ini menunjukkan periode waktu selama sebuah objek melakukan sebuah event.

Tabel 2.5 Komponen *Sequence Diagram*

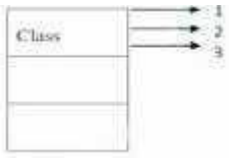


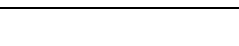

Simbol	Nama Elemen	Keterangan
	<i>Life Line</i>	Objek entitas, antar muka yang saling berinteraksi
	<i>Activation</i>	Menggambarkan hubungan antara objek dengan <i>message</i>
	<i>Message (call)</i>	Menggambarkan alur message yang merupakan kejadian objek pengirim <i>life line</i> ke objek penerima
	<i>Message (return)</i>	Menggambarkan alur pengambilan <i>message</i> ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.
	<i>Object</i>	Object adalah instance dari sebuah class yang dituliskan tersusun secara horizontal diikuti lifeline

2.9.4 Class Diagram

Class Diagram adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan saling berhubungan secara diagram ini menggambarkan alur struktur statis dari sebuah sistem. Karena itu *Class Diagram* merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML.

INFORMATICS & BUSINESS INSTITUTE DARMAJAYA
BANDAR LAMPUNG

Tabel 2.6 Komponen *Class Diagram* 2019

Simbol	Nama Elemen	Keterangan
	<i>Class</i>	Simbol untuk membangun sebuah pemrograman dengan objek. Terdiri 3 bagian, bagian atas adalah nama kelas, bagian tengah adalah atribut dan bagian bawah adalah metode dari kelas tersebut.
	Generalisasi	Simbol yang menandakan adanya generalisasi dari kelas input untuk menghasilkan data yang dibutuhkan.
	Asosiasi berarah / directed association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Kebergantungan /dependency	Relasi antar kelas dengan makna ketergantungan antar kelas.
	Agresi /aggregation	Relasi antar kelas dengan makna semua – bagian (<i>whole-part</i>).

2.10 Black-Box Testing (Pengujian Kotak Hitam)

Menurut (Rosa, 2016), *Black-box testing* yaitu perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan.