BAB II
TINJAUAN PUSTAKA

2.1 Kamus

Kamus merupkan sejenis buku rujukan yang menerangkan makna kata-kata. Kamus 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.(Darnila et al.,2018)

2.2 Bahasa Sansekerta

Bahasa Sanskerta adalah salah satu rumpundalam keluarga bahasa Proto Indo Eropa yangbanyak melahirkan bahasa-bahasa di Eropa. Sejalandengan perdagangan dan persebaran agama Hindudan Budha di Asia Tenggara, termasuk nusantara,bahasa Sanskerta yang digunakan untuk menulisWeda menjadi lebih dominan berpengaruh terhadapbahasa Jawa, Bali, dan Melayu Kuno. BahasaSanskerta digunakan sebagai bahasa kesusasteraanIndia yang hanya dikuasai oleh para cerdik pandaidan para pendeta (Wurianto, 2015).

2.3 Sistem Operasi

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* (Abas ali and Dony 2005). 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

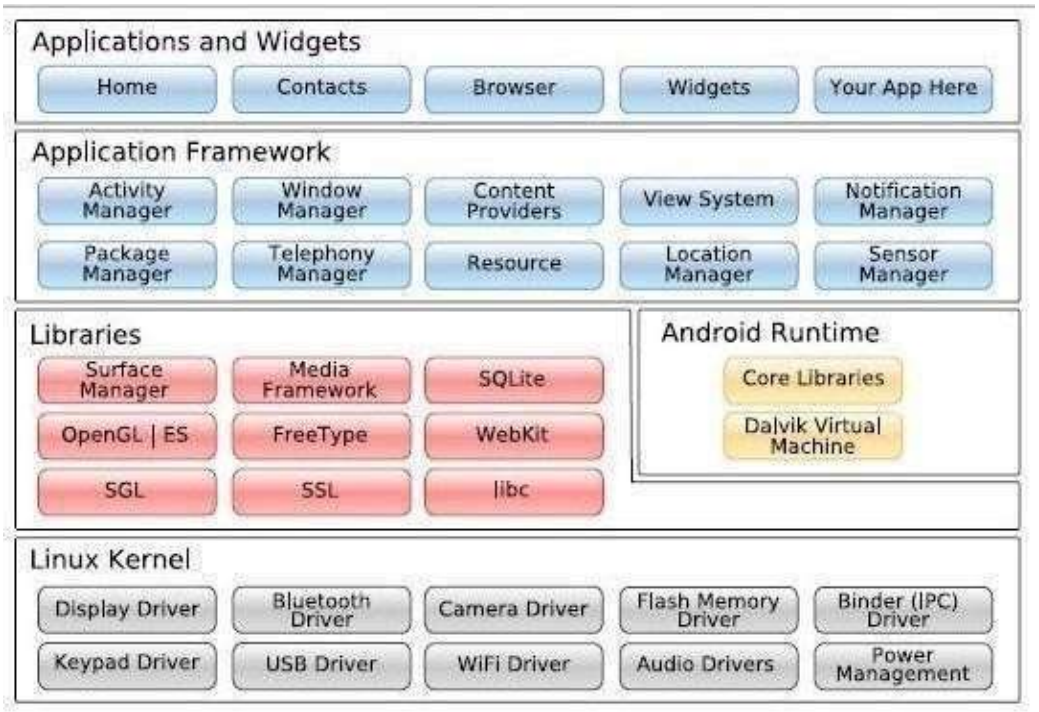
Sistem Operasi tersebut.

2.4 Android

Android merupakan suatu sistem operasi yang digunakan perangkat *mobile* yang dengan berbasis *linux* serta mencakup *middleware*, aplikasi dan sistem operasi (N. Safaat 2012). Sebagai *open source* dan bebas dalam memodifikasi, di dalam *Android* tidak memiliki ketentuan yang tetap di dalam konfigurasi *hardware* dan *software*.

2.5 Arsitektur Android

Secara garis besar arsitektur *Android* dapat dijelaskan dan digambarkan sepertipada gambar 2.1 (N. Safaat 2012).



Gambar 2. 1 Arsitektur Android

1. *Application* dan *Widgets*

   Application dan *Widgets* merupakan *layer* pengguna yang berhubungan dengan aplikasi saja, download aplikasi melakukan instalasi dan menjalankan aplikasi tersebut.

2. *Application Frameworks*

   Android memberi atau menawarkan kepada pengembang kemampuan untuk membangun aplikasi yang inovatif dan bagus, *Android* adalah "*Open Development Platform*" Pengembang bebas untuk mengatur *alarm*, *background service*, mengakses perangkat keras, informasi *resource* akses, *notification*, menambah status, dan sebagainya. Akses Pengembang memiliki akses penuh API *framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dapat dengan mudah menggunakan kembali *reuse* (komponen yang sudah digunakan). Disimpulkan bahwa *Application Framework* merupakan sebuah lapisan bagi pengembang aplikasi dalam mengembangkan aplikasi yang sedang dijalankan pada sistem operasi *Android*, karena pada lapisan ini aplikasi bisa dibuat dan dirancang, seperti penyedia konten berupa telepon dan SMS. Komponen-komponen yang termasuk ke dalam *Application Frameworks* adalah sebagai berikut:

   a. *Views*
   b. *Resource manager*
   c. *Notification manager*
   d. *Content manager*
   e. *Activity manager*
   f. *Libraries*

   Ini merupakan *layer* dimana fitur-fitur *libraries Android* berada, biasanya para pembuat aplikasi mengakses untuk menjalankan aplikasinya *libraries Layer* ini berjalan di atas Kernel, yang meliputi berbagai *library* inti C/C++, seperti :

a. *Libraries* untuk manajemen tampilan.
b. *Libraries* Graphics mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
c. *Libraries* media untuk pemutaran media *audio* dan *video*.
d. *Libraries* SQLite untuk dukungan *database*.
e. *Libraries* Live*Web*core mencakup moderen *web browser* dengan engine *embedded web view*.
f. *Libraries* 3D yang mencakup implementasi OpenGL ES1.0 API's.
g. *Libraries* SSL dan *Web*Kit terintegrasi dengan *web browser* dan *security*.

3. *Android Run Time*

Layer ini yang membuat aplikasi *Android* dapat dijalankan, dimana di dalam prosesnya memakai implementasi linux. Di dalam *Android* run time dipecah menjadi 2 bagian, sebagai berikut:

a. *Core Libraries, Android* Aplikasi dibentuk dalam bahasa *Java*, sedangkan DVM selaku *virtual* mesin, sehingga dibutuhkan suatu *libraries* yang berfungsi untuk menterjemahkan bahasa *Java*/ C yang ditangani oleh *Core Libraries*.
b. Dalvik *Virtual Machine*: *Virtual* mesin berbasis *register* yang dimaksimalkan untuk melaksanakan fungsi- fungsi secara efisien, di mana ialah pengembangan yang sanggup membuat *Linux* Kernel buat melaksanakan threading serta manajemen tingkatan rendah.

4. *Linux* Kernel

Linux kernel merupakan *layer* dimana inti dari sistem operasi dari *Android* itu terletak. Berisi file-file sistem yang mengendalikan *processing, drivers, resource, Memory*, serta sistem- sistem pembedahan *Android* yang lain. *Linux Kernel* yang digunakan *Android* merupakan *Linux Kernel release* 2.6 (N. Safaat 2012).

2.6 Aplikasi

Aplikasi merupakan sekelompok *atribut* yang terdiri dari beberapa *form, report* yang disusun sedemikian rupa sehingga dapat mengakses data (Rahman et al. 2015). 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. Mengkonsumsi Daya yang Rendah, Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop.

2.7 Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi *android* dan bersifat *open source* atau gratis (Juansyah 2015). *Android* Studio mengantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi *Android*. (Yudha Yudhanto and Ardhi Wijayanto 2018) *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 :

a. Projek berbasis pada *Gradle Build*
b. *Refactory* dan pembenahan *bug* yang cepat
c. *Tools* baru yang bernama "*Lint*" dikalim dapat memonitor kecepatan,kegunaan, serta kompetibelitas aplikasi dengan cepat.
d. Mendukung *Proguard And App-signing* untuk keamanan.

e. Memiliki GUI aplikasi *Android* lebih mudah. didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

2.8 Java

Java adalah salah satu bahasa pemrograman komputer yang memungkinkan programmer dapat memberi instruksi pada komputer. Java juga didefinisikan sebagai suatu sekumpulan teknologi yang berfungsi untuk membuat dan menjalankan perangkat lunak pada komputer. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, yang saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi berbasis java umumnya dikompilasi kedalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM) (Diana Okta Pugas, Maman Somantri, 2011).

2.9 Database

Sistem basis data merupakan sistem yang terdiri dari atas kumpulan tabel yangsaling berhubungan (dalam sebuah basis data disebuahsistem komputer) dan sekumpulan program (yang biasa disebut DBMS atau *DataBase Management System*) yang memungkinkan beberapa pemakaidan/atau program lain untuk mengakses dan memanipulasi tabel-tabel data tersebut.

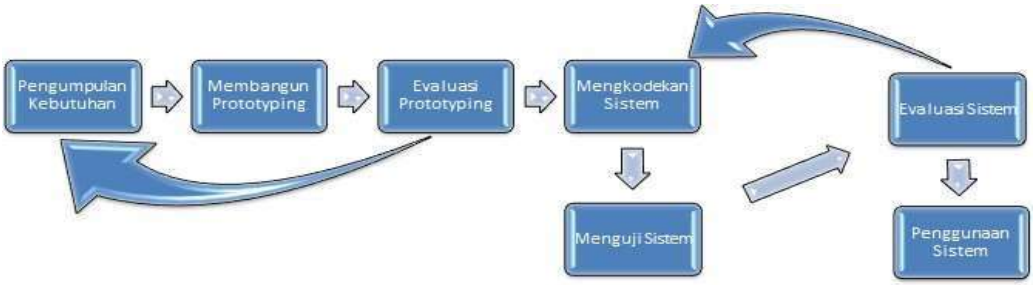
2.10 SQLite

SQLite adalah suatu library yang menerapkan mesin database self-contained, serverless, zero-configuration, dan transactional. Self-contained berati SQLite membutuhkan sedikit sekali dukungan dari library eksternal atau dari sistem operasi. Serverless berati SQLite dalam mengakses database baik itu read atau write dapat secara langsung dari file database tanpa melalui proses server dan tidak mendukung pengaksesan secara remote (artinya database SQLite bisa dikendalikan dari jarak jauh dengan adanya jaringan komputer (“Computer Network”), baik melalui jaringan lokal (intranet) atau internet), dimana ke-banyakan mesin SQL database diterapkan

sebagaiproses server yang terpisah (Setiyadi A., Harihayati T. 2015).

2.11 Prototype

Prototyping perangkat lunak (*software prototype*) atau siklus hidup menggunakan *prototyping* (*life cycle using prototyping*) yaitu salah satu metode siklushidup sistem yang didasarkan pada konsep model bekerja (*working model*). Tujuannya untuk mengembangkan model menjadi sistem *final*. Artinya sistem akan dikembangkan lebih cepat dari pada metde tradisional dan biaya menjadi lebih rendah. Ada banyak cara untuk membuat sebuah *prototype*, begitu juga dengan penggunaannya. Ciri khas dari metode ini adalah pengembang sistem, klien dan pengguna daat melihat dan melakukan eksperimen dengan bagian dari sistem komputer sejak awal prose pengembangan yang dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Metode Prototype

1. Pengumpulan kebutuhan

   Pelanggan dan pengambang bersama-sama mendifinisikan format seluruh perangkat luna dengan mengidentifikasikan semua kebutuhan dan garis besar sistem yang akan dibuat.

2. Membangun *prototype*

   Membangun *prototype* dengan membuat perancangan sementara untuk penyajian kepada pelanggan.

3. Evaluasi *prototype*

   Evaluasi ini dilakukan oleh pelanggan apakah *prototype* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4

akan diambil. Jika tidak *prototype* direvisi dengan mengulang langkah 1, 2 dan3.

4. Mengkodekan sistem

   Dalam taham ini, *prototype* yang sudah disepakati akan dimasukan ke dalam bahasa pemrograman yang sesuai.

5. Menguji sistem

   Setelah sistem mejadi perangkat lunak yang siap pakai, terlebih dahulu sistem diuji sebelum digunakan. Pengujian ini dulakukan dengan *white box* dan *blackbox* berbasis path pengujian arsitektur dan lain-lain.

6. Evaluasi sistem

   Pelanggan mengevaluasi apakah sistem yang sudah jadi sesuai dengan yang diharapkan. Jika iya, maka langkah 7 dilakukan, jika tidak, maka langkah 4 dan5 dilakukan.

7. Menggunakan sistem

   Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

2.12 Sistem

Sistem berasal dari bahasa latin yaitu *systẽma* atau bahasa yunani *sustẽma* yangberarti suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Sistem jugamerupakan sebuah kesatuan bagian-bagian yang saling memiliki hubungan yang berbeda dalam suatu wilayah, serta memiliki item-item sebagai penggerak.

Sistem didefinisikan menjadi “Suatu sistem adalah suatu jaringan kerja dari prosedurprosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sistem menurut Marshall B. Romney dan Paul Jhon Steinbart, Sistem (system) adalah serangkaian dua atau lebih komponen yang saling terkait dan berinteraksi untuk mencapai tujuan. Berdasarkan definisi diatas sistem adalah kumpulan atau serangkaian komponen yang saling terikat untuk mencapai suatu tujuan tertentu.(Kunci, 2020)

Menurut Jerry Fitsgerald dan Warren D. Stalling, Jr mengatakan bahwa " sistem adalah kumpulan dari elemen-elemen yang berinteraksi serta jaringan kerja dari prosedur- prosedur yang saling berhubungan bersama-sama untuk melakukan sesuatukegiatan atau untuk menyelesaikan suatu sasaran tertentu" (Sudradjat, 2020). Menurut Kurnia Cahya Lestari dan Arni Muarifah Amri mengemukakan bahwa Sistem mengemukakan bahwa sistem adalah dua atau lebih komponen yang saling berhubungan dan berinteraksi membentuk kesatuan kelompok guna menghasilkan satu tujuan yangdiharapkan. Sistem adalah sebuah tatanan yang mempunyai tujuan untuk memenuhi suatu proses tertentu karena sistem memiliki satuan dan tugas khusus yang saling berhubungan.(Dewi, 2021)

2.13 Algoritma Boyer Moore

Algoritma boyer moore adalah dengan melakukan pencocokan dari paling kanan string yang dicari. Dengan menggunakan algoritma ini, secara rata-rata proses pencarian akan lebih cepat dibandingkan dengan proses pencarian lainnya. Ide dibalikalgoritma ini adalah bahwa dengan memulai pencocokkan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat. Algoritma boyer moore menggunakan metode pencocokan string dari kanan ke kiri yaitu men-scan karakter pattern dari kanan ke kiri dimulai dari karakter paling kanan. Algoritma BoyerMoore menggunakan dua fungsi shift yaitu good-suffix shift dan bad-character shift untuk mengambil langkah berikutnya setelah terjadi ketidakcocokan antara karakter pattern dan karakter teks yang dicocokkan(Rahmanita, 2014).

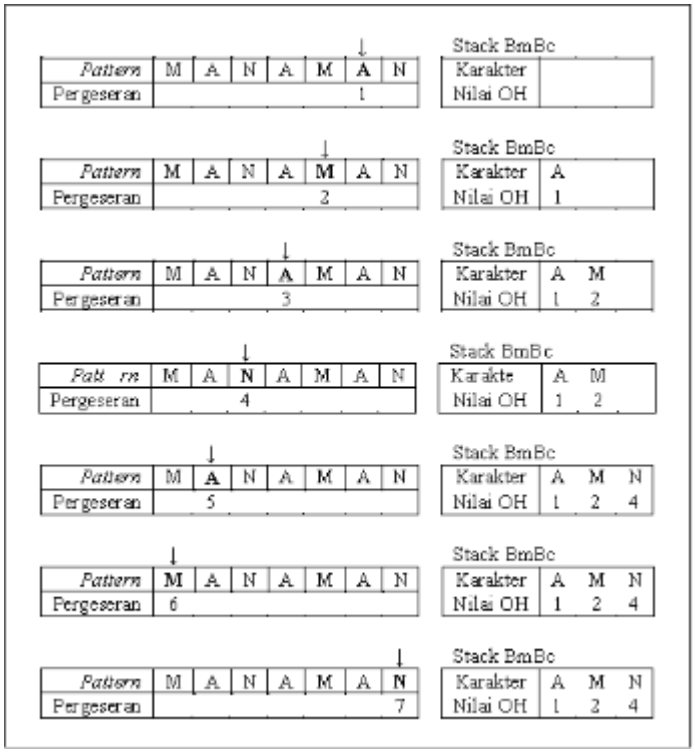
Cara kerja algoritma boyer moore adalah dengan menjalankan prosedur preBmBc danpreBmGs untuk mendapatkan inisialisasi (Borman, 2016).

1. Prosedur preBmBc

   Fungsi dari prosedur ini adalah untuk menentukan berapa besar pergeseran yang dibutuhkan untuk mencapai karakter tertentu pada pattern dari karakter pattern terakhir/terkanan. Hasil dari prosedur preBmBc disimpan pada tabel BmBc.

Contoh kasus :

Pattern : MANAMAN



Gambar 2. 3 Prosedur preBmBc

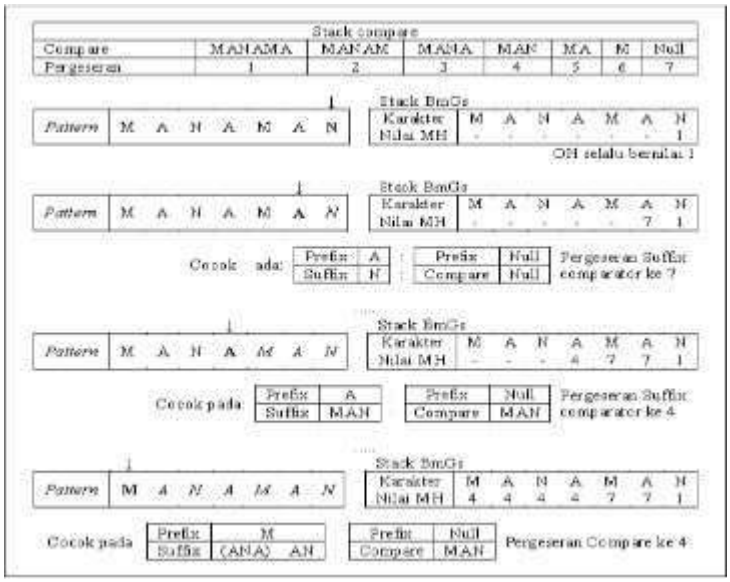
2. Prosedur preBmGs

   Sebelum menjalankan isi prosedur ini, prosedur suffix dijalankan terlebih dulu pada pattern. Fungsi dari prosedur suffix adalah memeriksa kecocokan sejumlah karakter yang dimulai dari karakter terakhir/terkanan dengan sejumlah karakter yang dimulai dari setiap karakter yang lebih kiri dari karakterterkanan tadi. Hasil dari prosedur suffix disimpan pada tabel suff. Jadi suff[i]

   mencatat panjang dari suffix yang cocok dengan segmen dari pattern yang diakhiri karakter ke-i.

   Contoh kasus :

   Pattern : MANAMAN



Gambar 2. 4 Prosedur preBmGs

2.14 Pengujian Sitem

White box Testing adalah salah satu cara untuk menguji suatu aplikasi atau *Software* dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Kalau modul yang telah dan sudah dihasilkan berupa output yang tidak sesuai dengan yang di harapkan maka akan dikompilasi ulang dan di cek kembali kode- kode tersebut hingga sesuai dengan yang diharapkan (Nidhra 2012).

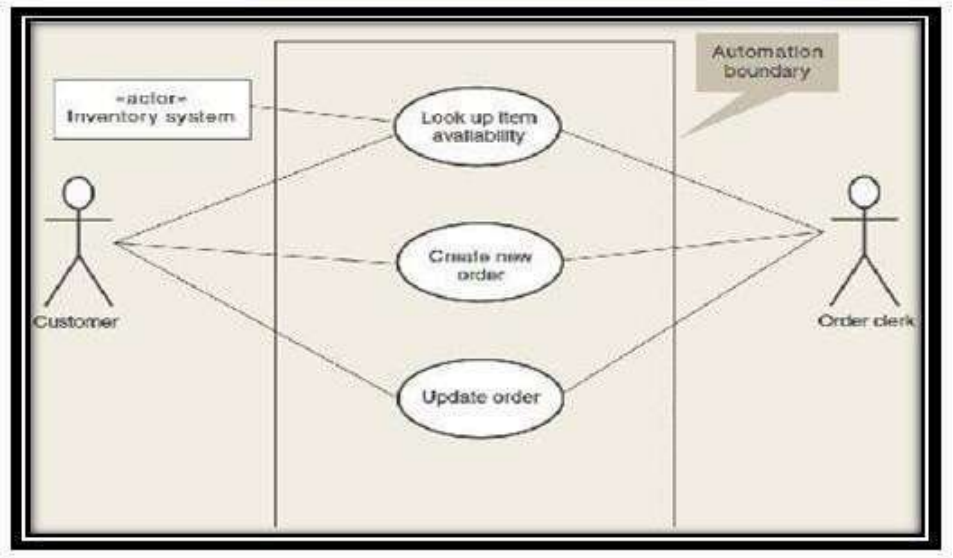
2.15 Unified Modeling Language (UML)

UML berkembang pesat pada akhir 1980 -an dan awal 1990-an UML lahir dari penggabungan banyak bahasa permodelan grafis yang berorientasikan objek. UML dibuat oleh James Rumbaugh, Ivar Jacobson dan Grady Booch. di bawah bendera

Rational Software Corp. UML memberika notasi-notasi yang dapat membantu memodelkan sistem dari berbagai perspektif. UML juga tidak hanya dapat digunakan dalam pemodelan perangkat lunak, hampir dalam semua bidang yang membutuhkan pemodelan (Fowler 2003).

1. Use Case Diagram

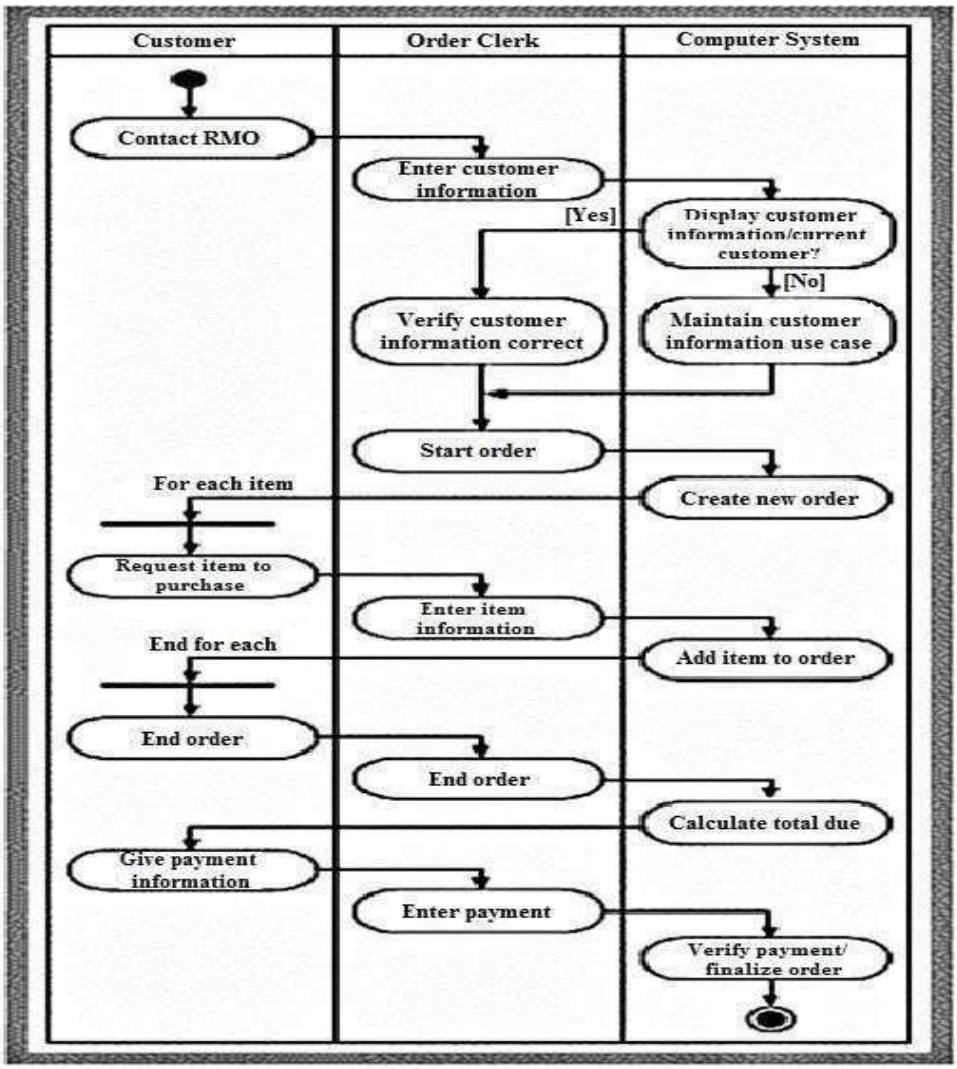
   Menyajikan interaksi antara *usecase* dan *actor* merupakan diagram *usecase*. Contoh diagram *usecase* disajikan pada Gambar 2.3 (John W. Satzinger 2016).



Gambar 2. 5 Use Case Diagram

2. Activity Diagram

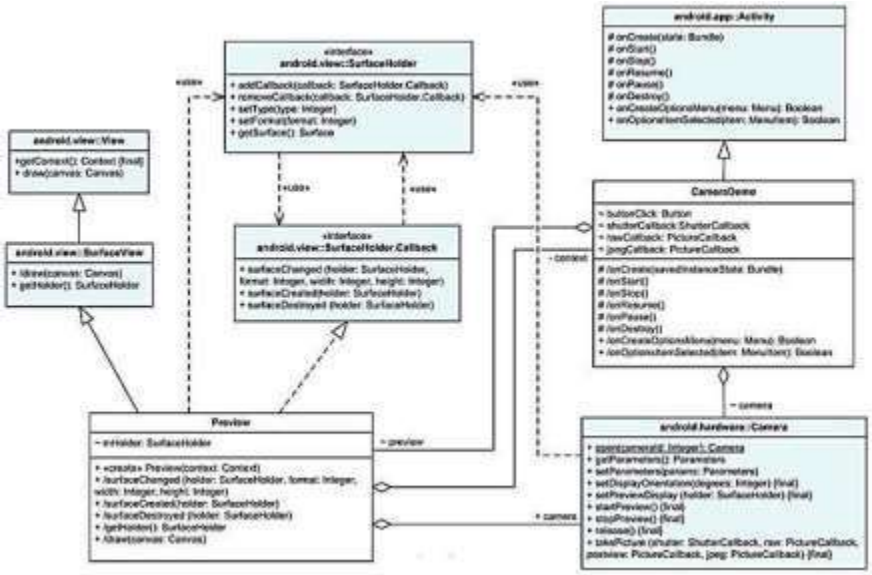
   Activity diagram ataupun Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Pada sesi pemodelan bisnis, diagram kegiatan bisa digunakan untuk menggambarkan aliran peristiwa (*flow of events*) dan membuktikan aliran kerja bisnis (*business work flow*) dalam *usecase*. Contoh diagram aktivitas disajikan pada Gambar 2.4 (John W. Satzinger 2016).



Gambar 2. 6 Activity Diagram

3. Class Diagram

   Class merupakan suatu spesifikasi yang menciptakan suatu objek serta menampilkan inti dari pengembangan serta desain berorientasi objek. *Class* menggambarkan kondisi (atribut/properti) sesuatu sistem, sekaligus menawarkan layanan untuk memanipulasi kondisi tersebut (tata cara/fungsi). *Class diagram* menggambarkan *package* serta objek beserta ikatan satu sama lain semacam pewarisan, struktur serta deskripsi class, asosiasi, serta lain- lain (Fowler 2003). Bentuk dari *Class diagram* dapat terlihat pada Gambar 2.5.



Gambar 2. 7 Class Diagram

Class diagram memiliki tiga area pokok utama yaitu:

a. Attribut

b. Operations

c. Class name

Pada UML diagram, *Class diagram* digambarkan menggunakan segi empat yang dibagi menjadi beberapa bagian. Bagian yang tengah merupakan struktur dari *Class diagram* (atribut), Bagian atas merupakan nama dari *Class diagram* (*class name*), dan bagian bawah merupakan sifat dari *class* (metode/operasi). Metode dan Atribut dapat memiliki salah satu sifat berikut (Fowler 2003).

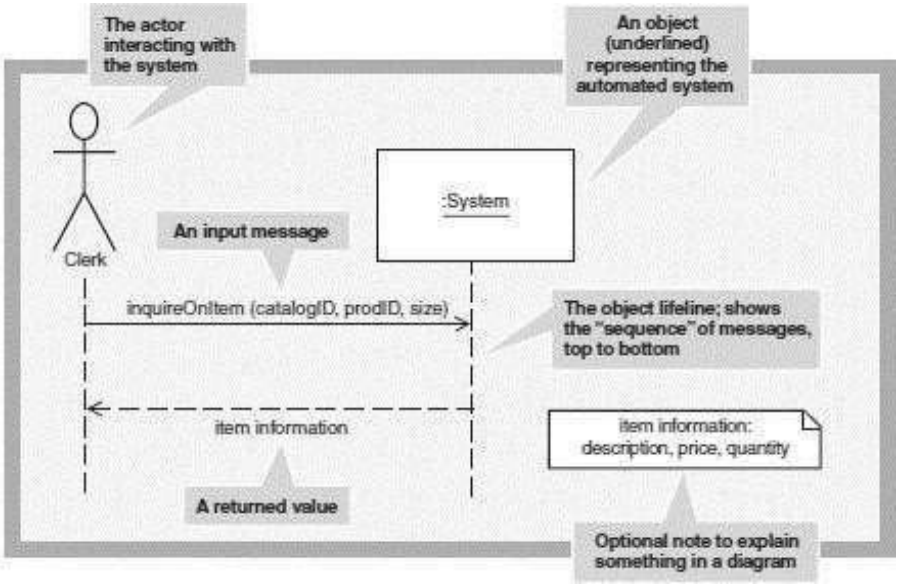
a. *Public*, bisa di panggil dari *class* lain. Antar *Class* Asosiasi, adalah hubungan statis antar *class*. Umumnya digambarkan *class* yang harus mengetahui eksistensi lain class atau class yang memiliki atribut berupa lain *class*. (Fowler 2003).

b. *Private*, hanya dapat di panggil di *class* yang bersangkutan.

c. *Protected*, hanya bisa dipanggil oleh *class* lain yang di warisinya dan *class* yang bersangkutan.

4. Sequence Diagram

Sequence diagram atau Diagram sekuensial biasa digunakan untuk menunjukan aliran fungsionalitas dalam *usecase*. Contoh pada *usecase* "menarik uang" mempunyai beberapa kemungkinan, seperti percobaan penarikan uang tanpa kecukupan ketersediaan dana, penarikan uang secara normal, penarikan dengan penggunaan PIN yang salah, dan lainnya (Sholiq 2006). Contoh diagram sekuensial disajikan pada Gambar 2.6 (John W. Satzinger 2016).



Gambar 2. 8 Sequence Diagram

2.16 Penelitian Relevan

Adapaun referensi yang peneliti gunakan sebagai rujukan pada penelitian kaliini adalah sebagai berikut :

Tabel 2. 1 Penelitian yang Relevan

No	Nama	Judul	Uraian	Keterangan
1	Agung Prayitno, Asahar Johar, Yudi Setiawan	Implementasi Algoritma Turbo Boyer Moore Pada Aplikasi Kamus Istilah Biologi Berbasis Android	Jurnal Rekursif, Vol. 6 No. 1 Maret 2018	Penelitian ini telah menghasilkan Aplikasi Kamus Istilah Biologi yang dapat berjalan dengan mengimplementasikan Algoritma Turbo Boyer Moore pada smartphone Android sebagai pencarian arti istilah Biologi bagi pengguna Android.
2	Yudika Ardi, Desi Andreswari, Yudi Setiawan	Rancang Bangun Aplikasi Kamus Istilah Kedokteran Dengan Menggunakan Algoritma Boyer-Moore Berbasis Android	Jurnal Rekursif, Vol. 5 No. 3 November 2017	Aplikasi kamus istilah kedokteran ini dapat menampilkan hasil pencarian istilah yang tersimpan di dalam database dengan menggunakan algoritma Boyer-Moore digunakan sebagai media untuk pencarian istilah kedokteran pada smartphone Android
3	Ni Made Suryati	Bahan Ajar Bahasa Sanskerta	Program Studi Sastra Bali Fakultas Ilmu Budaya Universitas Udayana Denpasar, 2016	Bahasa Sanskerta memiliki susunan yang lebih rapi dan terang, serta lebih mudah dipisah-pisahkan. Oleh karena itu pada abad XIX bahasa Sanskerta selalu dipelajari oleh ahli bahasa Eropa untuk pemahaman yang lebih jelas dan mendalam tentang bahasa Latin dan Yunani.

No	Nama	Judul	Uraian	Keterangan
4	Arif Budi Wurianto	Kata Serapan Bahasa Sanskerta Dalam Bahasa Indonesia	Jurnal Keilmuan Bahasa, Sastra, dan Pengajarannya, Oktober 2015, Volume 1, Nomor 2, hlm 125-134	Penelitian ini mendeskripsikan tentang linguistis kosakata bahasa Melayu yang diperoleh dari bahasa Sansekerta dan perubahan bahasa Sanskerta dalam bahasa Melayu dan bahasa Indonesia