BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berikut ini adalah beberapa penelitian yang dilakukan oleh peneliti-peneliti terdahulu. Penelitian berikut adalah penelitian yang berkaitan mengenai penelitian yang penulis buat :

Berdasarkan Penelitian yang dilakukan oleh Nurul Faizah Rozy dan Sholahuddin Alisyahbana mengenai Perencana Keuangan Pada *Platform*Berbasis *SmartphoneMultiplatform*Dengan *FrameworkPhonegap*, pada penelitian Nurul Faizah Rozy dan Sholahuddin Alisyahbana menyimpulkan pengelolaan keuangan diperlukan guna mengatur keuangan risiko kecelakaan, penyakit, kematian, dan tuntutan hukum; Mengurangi utang pribadi/keluarga; Membiayai keuangan bila hidup ini tidak lagi dalam rentang usia produktif – terkait dengan tingkat yang lebih tinggi harapan hidup rata rata di suatu di negara; Membayar biaya-biaya untuk membesarkan anak; Memberikan alokasi pendidikan bagi anak-anak ke ke perguruan tinggi; Membiayai pernikahan anak perempuan kita; Untuk membeli kendaraan; Untuk membeli rumah; Mampu menentukan gaya hidup yang kita inginkan saat pensiun; Membayar biaya biaya perawatan jangka panjang; dan Mewariskan kesejahteraan ke generasi selanjutnya (anak, cucu, dan lain-lain). Dalam perkembangan teknologi saat ini penggunaan *smartphone* berkembang pesat sehingga dibutuhkan aplikasi perencana keuangan berbasis *smartphone*.

Untuk itu, dibangunlah sebuah aplikasi perencana keuangan berbasis *smartphone* dengan menggunakan framework *Phonegap*.

Berdasarkan penelitian yang dilakukan oleh Meyta Nastiti dan Andi Sunyoto mengenai Perancangan Aplikasi Manajemen Keuangan Pribadi Berbasis *Android*, Nastiti dan Andi Sunyoto menyimpulkan bahwa Aplikasi Manajemen Keuangan Pribadi Berbasis *Android* ini dapat memberikan kemudahan dalam penganggaran, audit, manajemen, kontrol, deposito dana yang disimpan untuk keputusan keuangan secara efektif dan efisien. Perencanaan keuangan termasuk pendapatan, biaya, piutang, utang, dana darurat, rencana anggaran dan aset.

Penelitian yang dilakukan oleh Sarip Hidayatuloh dan Indah Sari Agustin mengenai Analisis dan Perancangan Sistem Informasi Pencatatan Keuangan Pada Koperasi Lancar Jaya, pada penelitian ini Sarip Hidayatuloh dan Indah Sari Agustin menggunakan sistem yang masih manual dan menggunakan *software* yaitu MS. Excel 2007.

II.2. Landasan Teori

II.2.1. Aplikasi

Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah Program yang dibuat oleh manusia yang berfungsi untuk menyelesaikan permasalahan-permasalahan masalah yang akan dihadapi. (Zulfauzi, 2015 : 57).

II.2.2. *Android*

Android adalah sebuah sistem operasi pada *smartphone* yang bersifat terbuka dan berbasis pada sistem operasi *Linux*. *Android* bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka.*Android* menyediakan *platform*terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak. Awalnya, *GoogleInc*. membeli *AndroidInc*., 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, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-mobile*, dan *Nvidia*. Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google*merilis kode–kode *Android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *standard* terbuka perangkat seluler. (Nazruddin Safaat 2012 : 1).

Pada tahun 2005, Andy Rubin dan Larry Page melakukan pertemuan di kantor *Google*, pertemuan tersebut bukan pertemuan pertama. Mereka telah berjumpa tiga tahun sebelumnya, ketika Andy Rubin akan meliris *Smartphone* yang dibuatnya. *Smartphone* tersebut diberi nama “Sidkick” yang memakai mesin pencari (*Search Engine*) default *google*. *Google* meminang *Android* pada bulan juli 2005, diestimasi harganya sekitar USD 50 juta. (Yuniar Supardi, 2017 : 1).

II.2.2.1. Mengenal *Android*

Android merupakan sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi (Yuniar Supardi, 2017 : 1).

Beberapa pengertian lain dari *Android*, yaitu:

1. Merupakan *platform* terbuka (*open source*) bagi para pengembang (*programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli *GoogleInc* dari *AndroidInc*.
3. Bukan bahasa pemograman, tetapi hanya menyediakan lingkungan hidup *atau run environment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem operasi yang kecil (Yuniar Supardi, 2017 : 2).

Untuk mengembangkan *Android*, dibentuk OHA (*Open Handset Alliance*), yaitu konsorsium dari 34 perusahaan peranti keras (*Hardware*), peranti lunak (*software*), dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*,*Qualcomm*, *T-mobile*, dan *Nvidia* (Yuniar Supardi, 2017 : 2).

Saat ini *Android* bersaing dengan *Apple* dalam sistem operasi untuk PC Tablet. Terdapat dua jenis *distributor* sistem operasi *Android*, Pertama adalah yang terdapat dukungan penuh *Google* atau GMS (*Google Mail Service*) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai OHD (*Open Handset Distribution*) (Yuniar Supardi, 2017 : 3).

II.2.2.2. Arsitektur *Android*

Secara garis besar, arsitektur *Android* terdiri atas *Applications*, *Applications Framework*, *Libraries*, *Android Run Time*, dan *Linux Kernel* (Yuniar Supardi, 2017 : 3).

1. *Applications* dan *widgets* merupakan *layer* (lapis) dimana kita berhubungan dengan aplikasi saja.
2. *Applications Framework* merupakan *Open Development Platfrom* yang ditawarkan *Android* untuk dapat dikembangkan guna membangun aplikasi. Pengembang memiliki akses penuh menuju API *Frameworks* seperti yang dilakukan oleh aplikasi katagori inti. Komponen-komponen yang termasuk didalam *Applications Framework* adalah *Views*, *Content Provider*, *Resource Manager*, *Notification Manager*, dan *Activity Manager*.
3. *Libraries Run Time* merupakan *layer* yang membuat aplikasi *Android* dapat dijalankan dimana dalam prosesnya menggunakan implementasi *Linux*.
4. *Linux Kernel* merupakan layer inti dari sistem operasi *Android* berada (Yuniar Supardi, 2017 : 3).

II.2.2.3. Struktur Aplikasi *Android*

Struktur Aplikasi *Android* atau *Fundamental* Aplikasi ditulis dalam bahasa pemograman *java*. Kode *java* dikompilasi bersama *resource file* yang dibutuhkan oleh Aplikasi. Prosesnya di *package* oleh *tools* yang dinamakan “apt tools” ke dalam paket *Android*, sehingga menghasilkan *file* berekstensi apk. *File* apk ini yang disebut dengan aplikasi dan nantinya dapat anda jalankan pada peralatan *mobile*(*device mobile*). (Yuniar Supardi, 2017 : 4).

Ada empat komponen dalam *Android*, yaitu:

1. *Activities* merupakan komponen untuk menyajikan tampilan pemakaian (*interface user*) kepada pengguna.
2. *Service* merupakan komponen yang tidak memiliki tampilan pemakai (*user interface*), tetapi *service* berjalan secara *backgrounds*.
3. *Broad Receiver* merupakan komponen yang berfungsi meneria dan beraksi untuk menyampaikan notifikasi.
4. *Content Provider* merupakan komponen yang membuat kumpulan apilkasi secara spesifik, sehingga bisa digunakan oleh aplikasi lain (Yuniar Supardi, 2017 : 5).

II.2.2.4. Versi *Android*

Banyak *Smartphone* dan Tablet yang menggunakan sistem operasi dengan versi yang berbeda. Semakin versi tinggi fiturnya, semakin canggih *smartphone* atau tablet tersebut. Telepon pertama yang memakai sistem operasi *Android* adalah *HTCDream* yang diliris pada tanggal 22 Oktober 2008 (Yuniar Supardi, 2017 : 5). Beberapa uraian versi *android*, yaitu:

Tabel II.1. Versi *Android*

No.	Nomor Versi	Nama Versi	Tanggal Rilis
1	(belum memakai)	*Android* Beta	5 November 2007
2	1.0	*Android* 1.0	23 September 2008
3	1.1	*Android* 1.1	9 Februari 2009
4	1.5	Cupcake	27 April 2009
5	1.6	Donut	15 September 2009
6	2.0	Eclair	26 Oktober 2009
7	2.0.1	Eclair	3 Desember 2009
8	2.1	Eclair	12 Januari 2010
9	2.2	Froyo	20 Mei 2010
10	2.2.1	Froyo	18 Januari 2011

11	2.2.2	Froyo	22 Januari 2011
12	2.2.3	Froyo	21 November 2011
13	2.3.	GingerBread	6 Desember 2010
14	2.3.3	GingerBread	9 Februari 2011
15	2.2.4	GingerBread	28 April 2011
16	2.3.5	GingerBread	25 Juli 2011
17	2.3.6	GingerBread	2 September 2011
18	2.3.7	GingerBread	21 September 2011
19	3.0	Honeycomb	22 Februari 2011
20	3.1	Honeycomb	10 Mei 2011
21	3.2	Honeycomb	15 Juli 2011
22	3.2.1	Honeycomb	20 September 2011
23	3.2.2	Honeycomb	30 Agustus 2011
24	3.2.4	Honeycomb	Desember 2011
25	3.2.6	Honeycomb	Februari 2012
26	4.0.1	ICS	18 Oktober 2011
27	4.0.2	ICS	28 November 2011
28	4.0.3	ICS	16 Desember 2011
29	4.0.4	ICS	29 Maret 2012
30	4.1	Jelly Bean	9 Juli 2012
31	4.4	Kitkat	31 Oktober 2013
32	5.0	Lolipop	12 November 2014
33	5.1	Lolipop	25 Juni 2014
34	6.0	MarshMallow	5 Oktober 2015
35	7.0	Nougat	22 Agustus 2016

(Sumber :Yuniar Supardi, 2017 : 5)

II.2.2.5. *Android*SDK

Android SDK adalah toolsAPI (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada*platformandroid* yang menggunakan bahasa pemrograman *Java*. *android* merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware*dan aplikasi kunci yang dirilis oleh *Google*. Saat ini disediakan *Android* SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mengembangkan aplikasi pada *platformAndroid*menggunakan bahasa pemrograman *Java*. Sebagai *platformandroid* aplikasi – *netral*, *android* memberi anda kesempatan untuk

membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *smartphone/smartphone*. (Nazruddin Safaat 2012 : 5).

II.2.2.6. ADT (*Android Development Tools*)

AndroidDevelopment Tools (ADT) adalah plug-inyang didesain untuk IDE *Eclipse*yang memberi kita kemudahan dalam mengembangkan aplikasi *android* dengan menggunakan IDE *Eclipse*. Dengan menggunakan ADT untuk *Eclipse*, ini akan memudahkan kita dalam membuat aplikasi *projectandroid*, membuat GUI aplikasi, dan menambahkan komponen - komponen yang lainnya. Selain itu kita juga dapat melakukan *running* aplikasi menggunakan *android* SDK melalui *Eclipse*. Dengan ADT kita juga dapat melakukan pembuatan *packageandroid* (.apk) yang digunakan untuk distribusi aplikasi *android* yang kita rancang. (Nazruddin Safaat 2012 : 6)

II.2.2.7. AVD (*Android* Virtual Device)

AVD merupakan *emulator* yang digunakan untuk menjalankan program aplikasi *Android* yang telah dirancang. AVD dapat dikonfigurasi agar dapat menjalankan berbagai macam versi *Android* yangtelah diinstal. (Nazruddin Safaat 2012 : 19).

II.2.3. *Java*

*Java*adalah bahasa pemrograman yang *multi platform*dan *multi device*. Sekali anda menuliskan sebuah program dengan menggunakan*Java.*Fungsionalitas dari *Java* ini dapat berjalan dengan *platform* sistem operasi yang berbeda karena sifatnya yang umum dan non-spesifik.

(Hardianto dan Handaga, 2015 : 4).

Java adalah bahasa pemrograman yang *multi platform*dan *multi device*. Sekali anda menuliskan sebuah program dengan menggunakan *Java*. Aplikasi dengan berbasis *Java* ini dikompulasikan ke dalam *p-code* dan bisa dijalankan dengan *Java Virtual Machine*. Fungsionalitas dari *Java* ini dapat berjalan dengan *platform* sistem operasi yang berbeda karena sifatnya yang umum dan non -spesifik. (Hardianto, 2015 : 3).

II.2.3.1,*Java Development Kit* (JDK)

Java adalah sebuah teknologi yang diperkenalkan oleh *SunMicrosysytems* pada pertengahan tahun 1990. Menurut definisi Sun, *Java* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *stand alone* ataupun pada lingkungan jaringan. Untuk membuat program *Java* dibutuhkan *compiler* dan *interpreter* untuk program *Java* berbentuk *JavaDevelopment kit* (JDK) yang diproduksi oleh Sun Microsystems. Sebelum memulai instalasi *Android* SDK, terlebih dahulu kita harus melakukan instalasi JDK di komputer. JDK yang kami gunakan untuk dapat mengompilasi aplikasi *android* yang kami rancang iniadalah *Java* SE *Development kit* 7. (DeCoster 2012).

II.2.4. Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak

untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

- Sistem versi berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- Alat pengujian dan kerangka kerja yang ekstensif
- Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- Dukungan C++ dan NDK
- Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

II.2.4.1. *Extensible Markup Language* (XML)

Extensible Markup Language (XML) adalah sebuah bahasa markah untuk mendeskripsikan data. XMLmerupakan turunan (*subset*) atau versi ringkas dari SGML (*Standart Generalized Markup Language*), sedangkan SGML merupakan sebuah standar ISOuntuk *format* dokumen. SGMLtidak berisi berupa *tag-tag* siap pakai seperti halnya bahasa HTML, melainkan berupa aturan-aturan standar dalam pembuatan *tag-tagformat* dokumen. SGML banyak dipakai untuk

mengelola dokumen dalam jumlah besar, frekuensi revisi tinggi dan dibutuhkan dalam beragam *format* tampilan. SGML jarang dipakai karena sangat rumit dan kompleks. XML dibuat dengan konsep yang lebih sederhana dan ringkas, tujuannya agar bisa dipakai sebagai aplikasi *desktop* dan jaringan internet. (Yenni dan Shamir, 2012 : 107).

II.2.5. SQLite

SQLite adalah sebuah *embeddeddatabase* yang sangat terkenal karena menggabungkan antarmuka SQL dengan memori yang sangat kecil dan kecepatan yang baik. (Murphy 2010 : 225).

SQLite adalah sebuah *open sourcedatabase* yang telah ada cukup lama, cukup stabil, dan sangat terkenal pada perangkat kecil, termasuk *Android*. (Gargenta 2011 : 119).

II.2.6. *Unified Modeling Language* (UML)

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

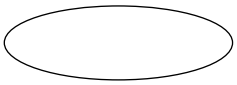
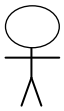
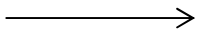
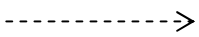
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Urva dan Siregar, 2015 : 93).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih *aktor* dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use casediagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan *aktor*, dan dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	*Aktor* adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan *aktor*, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa *aktor* berinteraksi dengan *use case*, tetapi tidak memiliki *control* terhadap *use case*.
	Asosiasi antara *aktor* dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara *aktor* dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila *aktor* berinteraksi secara pasif dengan sistem.
	*Include,* merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain,

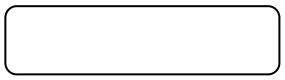
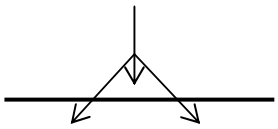
	contohnya adalah pemanggilan sebuah fungsi program.
<----------	*Extend,* merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Urva dan Siregar, 2015 : 94)

2. *Diagram* Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3di bawah ini:

Tabel II.3. Simbol *Activity Diagram*

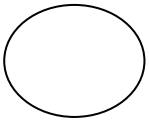
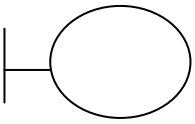
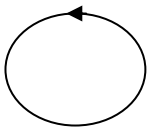
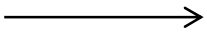
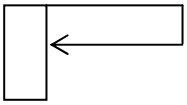
Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara pararel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points,* menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane*, untuk menunjukkan siapa melakukan apa.

(Sumber : Urva dan Siregar, 2015 : 94)

3. *Diagram* Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4di bawah ini :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	*EntityClass,* merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class,* berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih *aktor* dengan sistem, seperti tampilan *formentry* dan *form* cetak.
	*Control class,* suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive,* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Urva dan Siregar, 2015 : 95)

4. *Class Diagram* (*Diagram* Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggng jawab entitas yang menentukan perilaku sistem.*Class diagram* juga menunjukkan

atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility,* tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality* yang dapat dilihat pada tabel II.5di bawah ini:

Tabel II.5. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Urva dan Siregar, 2015 : 95)

II.2.7. Keuangan

Keuangan dalam KBBI (2008:1767) diartikan : (1) segala sesuatu yang bertalian dengan uang; (2) seluk beluk uang; (3) urusan uang; (4) keadaan uang. Contoh dalam kalimat: biaya rumah sakit tidak terjangkau oleh keuanganku. (artinya: kondisi uang/harta/kekayaanku tidak bisa menjangkau biaya rumah sakit)

Dalam Wikipedia bahasa Indonesia, Keuangan adalah mempelajari bagaimana individu, bisnis, dan organisasi meningkatkan, mengalokasi, dan menggunakan sumber daya moneter sejalan dengan waktu, dan juga menghitung risiko dalam menjalankan proyek mereka. Istilah keuangan dapat berarti: (1) Ilmu

keuangan dan asset lainnya; (2) Manajemen asset tersebut; (3) Menghitung dan mengatur risiko proyek.

Ridwan dan Inge (2003). Keuangan merupakan ilmu dan seni dalam mengelola uang yang mempengaruhi kehidupan setiap orang dan setiap organisasi. Keuangan berhubungan dengan proses, lembaga, pasar, dan instrumen yang terlibat dalam transfer uang diantara individu maupun antara bisnis dan pemerintah.

Referensi:

1. Pusat Bahasa Depdiknas, Kamus Bahasa Indonesia, Jakarta, 2008.
2. Sundjaja Ridwan S. & Barlian Inge, Manajemen Keuangan, edisi ke lima, Literata Lintas Media, Jakarta, 2003.
3. Wikipedia bahasa Indonesia tentang uang dan keuangan.