

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

(Febriyus, 2013). Dalam penelitiannya yang berjudul “Rancang Bangun Aplikasi *E-Reminder Services* (Layanan Peningat Elektronik) Untuk Kegiatan Dosen (Studi Kasus: Jurusan Teknik Informatika UIN Suska Riau)”. Masih kurang efisiennya aplikasi peningat untuk dosen karena membutuhkan peran aktif dari pengguna untuk memasukkan data yang akan diingatkan dan harus membuat peningat setiap kegiatan yang ingin dilakukan. Oleh karena itu, diperlukan sebuah aplikasi peningat yang dapat mengingatkan seluruh kegiatan tanpa memerlukan keterlibatan aktif dosen setiap hari dalam memasukkan data yang akan diingatkan. Dalam penelitian ini dibangun aplikasi untuk pengelolaan surat menyurat yang menggunakan fasilitas peningat untuk memberikan suatu peringatan atau peningat mengenai surat yang butuh balasan, selain itu juga dapat memberikan kemudahan kepada pengguna untuk dapat secara langsung melihat informasi data dari atribut surat secara cepat. Aplikasi ini dibuat menggunakan jaringan GSM untuk notifikasi SMS. Tujuan dari penelitian ini adalah membangun aplikasi layanan peningat elektronik (*e-reminder services*) kegiatan dosen berbasis SMS *gateway* dengan menggunakan pemrograman *PHP* dan penerapan fasilitas layanan *e-reminder services* kepada dosen.

(Ramadhan & Utomo, 2014). Dalam penelitiannya yang berjudul “Rancang Bangun Aplikasi Mobile Untuk Notifikasi Jadwal Kuliah Berbasis Android (Studi Kasus: STIMIK Provisi Semarang)”. Informasi jadwal kuliah, jadwal ujian akhir semester, jadwal ujian proposal, jadwal ujian skripsi jadwal pengambilan KRS dan jadwal pengambilan KHS di STIMIK ProVisi masih menggunakan papan pengumuman sehingga menimbulkan kesulitan bagi mahasiswa, terutama bagi mahasiswa dari luar kota dan yang berada pada tingkat akhir. Tujuan dalam penelitian ini adalah membangun sebuah aplikasi *mobile* yang dapat membantu mahasiswa dalam mendapatkan informasi penjadwalan. Metode perancangan aplikasi yang digunakan adalah metode SDLC dengan pendekatan *waterfall*.

Alasanya karena metode ini dianggap cocok dengan urutan pengembangan aplikasi yang akan dibuat.

(Putra, 2015). Dalam penelitiannya yang berjudul “Rancang Bangun Aplikasi Pengingat Jadwal dan Tugas Kuliah Berbasis Android”. Dalam kehidupan sehari-hari pada umumnya mahasiswa kesulitan dan sering lupa untuk mengatur dan menepati jadwal hariannya khususnya sering lupa jadwal ujian, untuk itu diperlukan sebuah aplikasi yang dapat membantu agenda mahasiswa sekaligus mengingatkan mahasiswa tersebut dimana pun dan kapan pun mereka berada bersama dengan smartphone mereka. Tujuan dari penelitian ini adalah untuk membantu agenda penjadwalan dan pengingat tugas mahasiswa. Metode yang digunakan dalam mengumpulkan suatu informasi sebagai pendukung, sekaligus pelengkap dalam pembuatan aplikasi ini adalah dengan *model waterfall*. Pembuatan aplikasi ini menggunakan *Eclipse Juno* dan *software editor* pendukung lainnya dan diprioritaskan berjalan pada android versi 4.2.1 (*jelly bean*).

(Findra, Theresia , & Yoris, 2016). Dalam penelitiannya yang berjudul “Rancang Bangun Aplikasi Pengingat Kegiatan Akademik Berbasis *Mobile*”. Tujuan dari penelitian ini adalah membangun aplikasi yang dapat digunakan sebagai pengingat jadwal kegiatan akademik dikampus UAJY. Penelitian ini terbagi kedalam dua tahap, pertama analisis dan perancangan aplikasi dan kedua pembangunan aplikasi. Metodologi dalam penelitian ini terbagi kedalam beberapa tahap, pertama yaitu studi literature yang dilakukan untuk mendapatkan informasi terkait dengan aplikasi yang akan dibangun. Kedua yaitu wawancara langsung kepada dosen dan mahasiswa UAJY untuk memperoleh data serta informasi mengenai proses mengingat jadwal akademik di UAJY. Ketiga yaitu analisis kebutuhan sistem, yang dilakukan untuk mengetahui kebutuhan sistem baik fungsional maupun nonfungsional dan akan menghasilkan spesifikasi sistem dan kebutuhan fungsionalitas yang harus tersedia dalam aplikasi. Keempat yaitu perancangan dan pembuatan mockup, yang dilakukan untuk menghasilkan rancangan aplikasi berdasarkan hasil analisis pada tahap sebelumnya. Kelima yaitu pembangunan aplikasi, yang dilakukan untuk menghasilkan aplikasi secara

keseluruhan. Keenam pengujian aplikasi, yang dilakukan untuk mengetahui kelebihan dan kekurangan aplikasi baik dari aspek fungsionalitas maupun *user acceptance*.

Berdasarkan empat penelitian yang menjadi tinjauan dalam melakukan penelitian, peneliti akan membangun suatu sistem dimana aplikasi berbasis android yang nantinya akan berguna untuk para mahasiswa dalam mendapatkan informasi penjadwalan acara melalui aplikasi berbasis android.

Adapun perbedaan pada penelitian ini yaitu pada penelitian sebelumnya aplikasi hanya dapat menampilkan informasi-informasi sedangkan pada penelitian ini menampilkan informasi-informasi penjadwalan acara dengan notifikasi setiap acara akan dimulai dan ditambahkan fitur komentar untuk setiap acara yang dihadiri.

2.2 Landasan Teori

Untuk mendukung hasil penelitian diperlukan suatu konsep dalam merumuskan definisi-definisi yang menunjang kegiatan penelitian baik teori dasar maupun teori umum.

2.2.1 Penjadwalan

(Pengertian Penjadwalan Menurut Para Ahli, n.d.) Jadwal menurut kamus besar bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan ke dalam jadwal. Menurut Chambers (1995:22) menyatakan bahwa jadwal didefinisikan sebagai sesuatu yang menjelaskan di mana dan kapan orang-orang dan sumber daya berada pada suatu waktu.

2.2.2 Aplikasi

(Hasugian, 2014) Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja *to apply* yang dalam bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu subkelas perangkat lunak komputer yang

menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Contoh utama perangkat lunak aplikasi adalah program pengolah kata, lembar kerja, dan pemutar media. Kumpulan aplikasi komputer yang digabung menjadi suatu paket biasanya disebut paket atau suite aplikasi (*application suite*). Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Umumnya aplikasi-aplikasi tersebut memiliki kemampuan untuk saling berinteraksi sehingga menguntungkan pemakai. Contohnya, suatu lembar kerja dapat dimasukkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan

Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*.

Pengertian aplikasi menurut para ahli adalah sebagai berikut :

- a. Menurut Jogiyanto (1999:12) adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output*.
- b. Menurut Kamus Kamus Besar Bahasa Indonesia (1998 : 52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu

program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

- c. Menurut Rachmad Hakim S, Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows &, permainan (game), dan sebagainya.
- d. Menurut Harip Santoso, Aplikasi adalah suatu kelompok file (form, class, rePort) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi payroll, aplikasi fixed asset, dan lain-lain.

2.2.3 Aplikasi Mobile

(Romdoni, 2010) Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon mobile berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi.

Sistem Aplikasi mobile merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain tanpa terjadipemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti telepon seluler dan tab.

2.2.4 Android

(Pengertian Definisi Android, 2017) Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. 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, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilis 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 standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Kerjasama dengan Android Inc. Pada Juli 2005, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

2.2.5 Android Studio

(Jadibaru, Pengenalan Android Studio, 2015) Android Studio adalah sebuah IDE untuk *Android Development* yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android.

Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment. Fitur-fitur lainnya adalah sebagai berikut :

- Menggunakan *Gradle-based build system* yang fleksibel.
- Bisa mem-build multiple APK .

- Template support untuk Google Services dan berbagai macam tipe perangkat.
- *Layout* editor yang lebih bagus.
- *Built-in support* untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
- Import library langsung dari Maven repository

2.2.7 Bahasa Pemrograman Java

(Vicky, Pengertian Pemrograman Java - Kelebihan dan Kekurangan, 2012)

Java adalah bahasa pemrograman yang multi *platform* dan *multi device*. Sekali anda menuliskan sebuah program dengan menggunakan Java, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang support Java, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. 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.

Kelebihan *java* yang pertama tentu saja *multiplatform*. Java dapat dijalankan dalam beberapa *platform* komputer dan sistem operasi yang berbeda. Hal ini sesuai dengan slogannya yang sudah dibahas sebelumnya. Yang kedua adalah OOP atau *Object Oriented Programming*. Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam Java. Hal ini akan memudahkan pemrograman menjadi lebih mudah. Kelengkapan *library* semakin beragam jika ditambah dengan karya komunitas Java.

Setiap hal pasti memiliki kelebihan dan kekurangan.. Beberapa hal harus disesuaikan jika dijalankan pada *platform* yang berbeda. Misalnya untuk J2SE dengan platform SWT-AWT bridge tidak dapat berfungsi di Mac OS X. Kekurangan lainnya adalah kemudahan aplikasi Java didekompilasi. Dekompilasi adalah suatu proses membalikkan sebuah aplikasi menjadi kode sumbernya. Hal ini memungkinkan terjadi pada Java karena berupa *bytecode* yang menyimpan bahasa tingkat tinggi. Hal ini terjadi pula pada platform .NET dari Microsoft sehingga

program yang dihasilkan mudah dibajak kodenya karena sulit untuk disembunyikan.

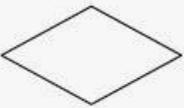
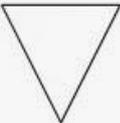
Kekurangan yang lain adalah penggunaan memori yang cukup banyak, lebih besar daripada bahasa tingkat tinggi sebelum generasi Java. Namun hal ini memang sesuai dengan fitur beragam yang dimiliki oleh Java. Masalah memori ini juga tidak dialami oleh semua pengguna aplikasi Java. Mereka yang sudah menggunakan perangkat keras dengan teknologi terbaru tidak merasakan kelambatan dan konsumsi memori Java yang tinggi. Lain halnya dengan mereka yang menggunakan teknologi lama atau komputer yang sudah berumur tua lebih dari empat tahun akan merasakan adanya kelambatan. Namun apapun kelemahan yang dimiliki Java, faktanya adalah Java merupakan bahasa pemrograman yang populer dan digunakan di seluruh dunia saat ini.

2.2.7 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 – Desember 1999 (Saat ini sudah terbit edisi ke-5 – Juni 2011). JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

2.2.8 Flowchart

Flowchart adalah bagan – bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma. Tujuan membuat *Flowchart* adalah menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapih dan jelas (Dwiky Andika). Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu flowchart pada gambar 2.1

	Simbol <i>process</i> , yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
	Simbol <i>manual</i> , yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
	Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak
	Simbol <i>predefined process</i> , yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
	Simbol <i>terminal</i> , yaitu menyatakan permulaan atau akhir suatu program
	Simbol <i>keying operation</i> , Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
	Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
	Simbol <i>manual input</i> , memasukkan data secara manual dengan menggunakan online keyboard

Gambar 2.1 Simbol *Flowchart*

2.2.9 *Unified Modeling Language (UML)*

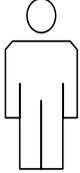
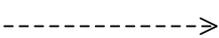
UML yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek atau sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software (Sora N, 2015).

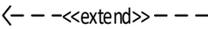
UML yang biasa digunakan selama kegiatan antara lain:

a. Use Case Diagram

Use case diagram yaitu salah satu jenis diagram pada *UML* yang menggambarkan interaksi antara sistem dan aktor, use case diagram juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya. Berikut simbol-simbol dalam *use case* diagram dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol *Use Case*

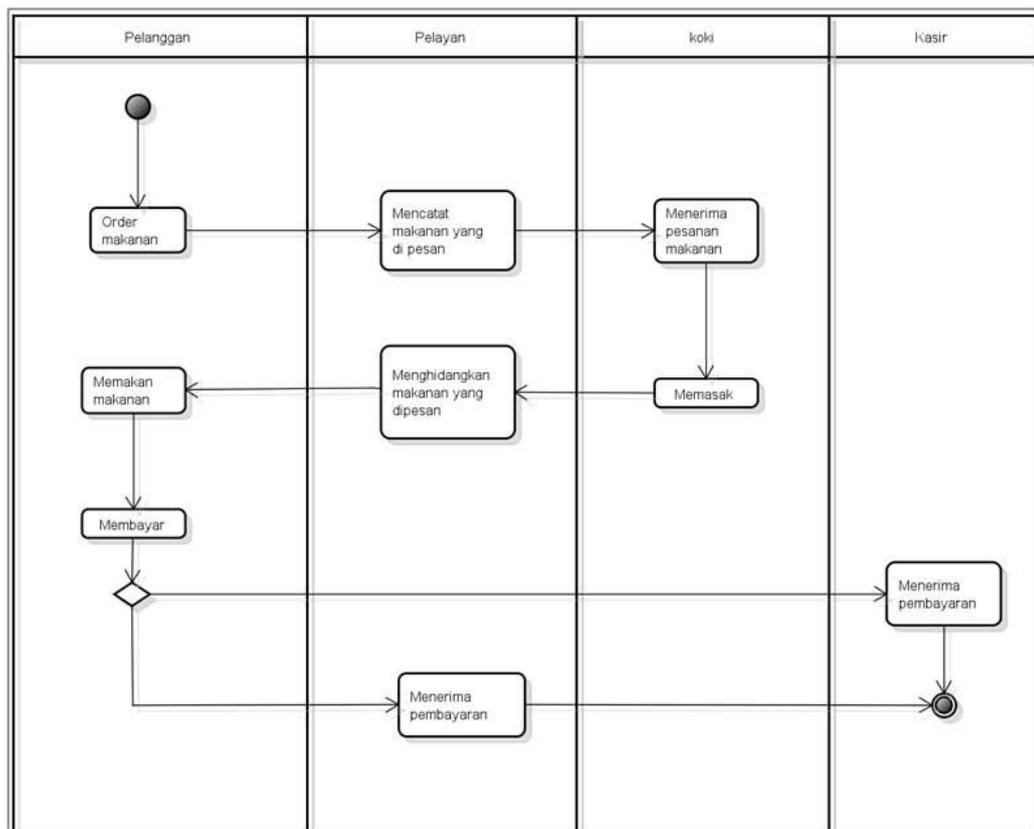
No.	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri (<i>independent</i>).
3.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.

No.	Gambar	Nama	Keterangan
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
8.		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Activity Diagram*

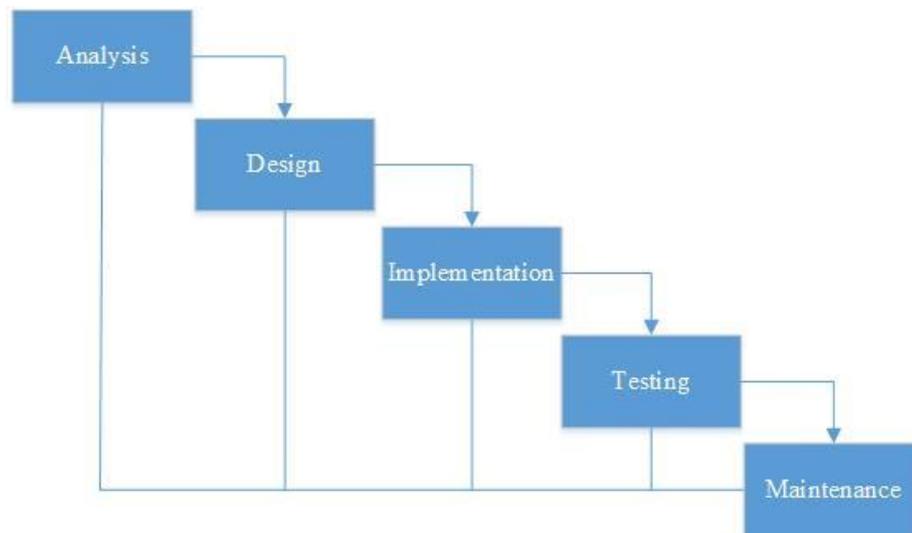
Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada *UML* yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem. Berikut contoh *activity diagram* pada tabel 2.2.

Tabel 2.2 Contoh *Activity Diagram*



2.2.11 Metode *Waterfall*

Metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan atau pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2010).



Gambar 2.2 Metode *Waterfall*

Tahapan metode waterfall, sebagai berikut:

a. Requirement Analysis

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

b. System Design

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

c. Implementation

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

d. Integration & Testing

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

e. Operation & Maintenance

Tahap akhir dalam model waterfall. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.2.12 Black Box Testing

Menurut Rex Black, *black box testing* adalah suatu metode pengujian dimana *tester* hanya fokus pada apa yang seharusnya dilakukan oleh sistem (Black, 2009, p. 3). Sebuah tes dapat dikatakan berhasil ketika sebuah sistem dapat memproses data dan hasil yang ada sesuai dengan apa yang diharapkan. Ketika menggunakan metode *black box*, *tester* tidak perlu mengetahui bagaimana struktur dan desain data yang ada di dalam sistem. Mereka hanya melihat apakah sistem terjadi *bugs* atau tidak (Tim Riley, 2010, p. 270).