



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

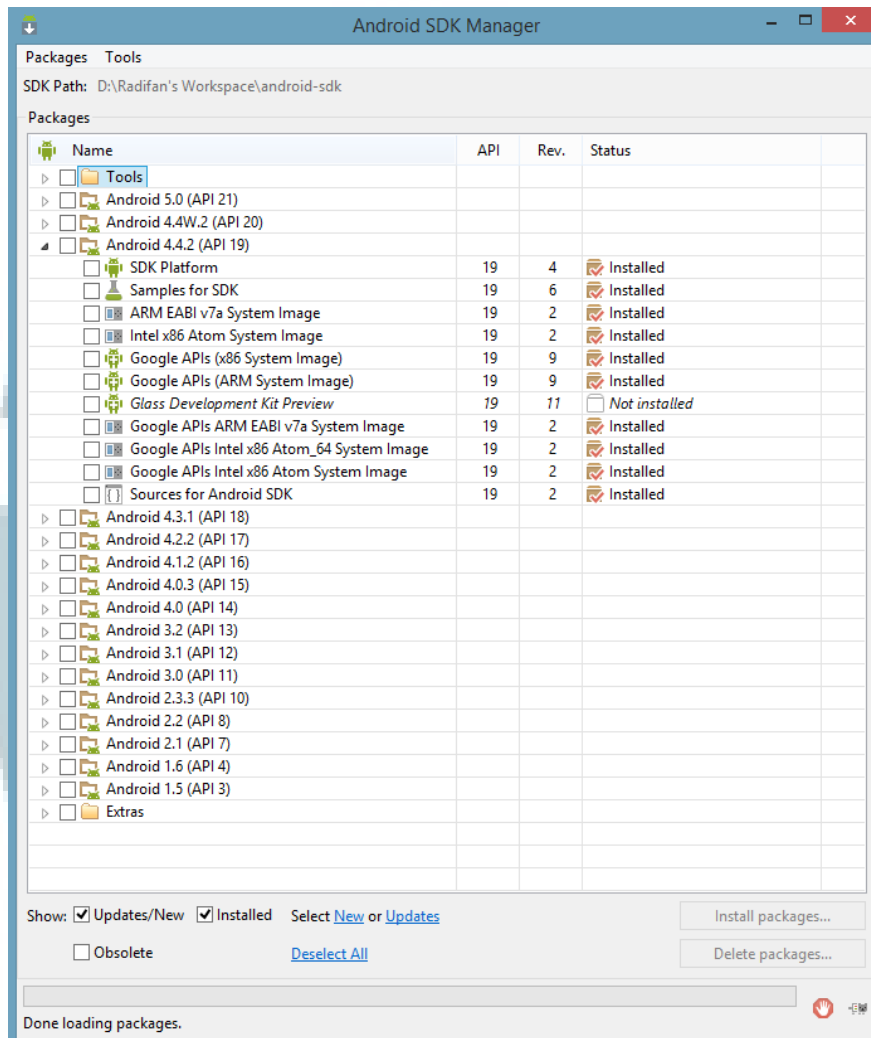
LANDASAN TEORI

2.1 Android

Android merupakan merupakan suatu sistem operasi berbasis Linux yang dirancang untuk perangkat seluler berlayar sentuh seperti smartphone dan tablet computer. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian dibeli oleh Google pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, yang merupakan konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

2.2.1 Android SDK

SDK merupakan singkatan dari Software Development Kit, yang menurut Nazrudin Safaat H [3] merupakan suatu tools berbasis API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform Android yang menggunakan bahasa pemrograman Java..



Gambar 2.1 Android SDK

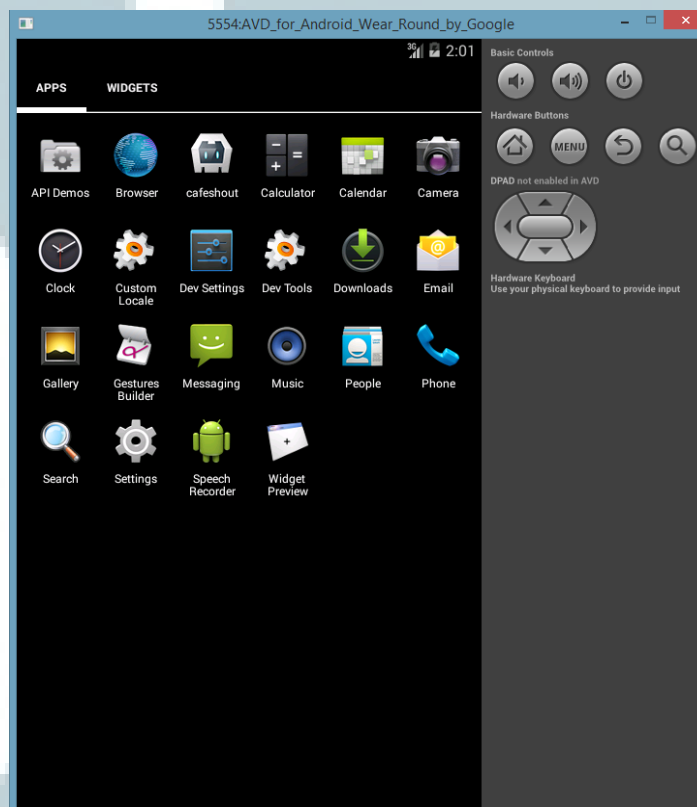
2.2.2 Android ADT

ADT Merupakan singkatan dari *Android Development Tools*, yang menurut Nazrudin Safaat H [3] merupakan suatu plug-in yang didesain untuk Eclipse yang dapat memudahkan kita dalam mengembangkan aplikasi Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya. ADT juga berfungsi untuk melakukan pembuatan package

Android (.apk) yang nantinya digunakan untuk mendistribusikan aplikasi Android yang telah kita rancang.

2.2.3 *Android AVD*

AVD merupakan singkatan dari *Android Virtual Device*, menurut Safaat [3] AVD merupakan suatu emulator yang digunakan untuk menjalankan program aplikasi Android yang telah dirancang. AVD dapat dikonfigurasi agar dapat menjalankan berbagai macam versi Android yang telah diinstall sebelumnya.

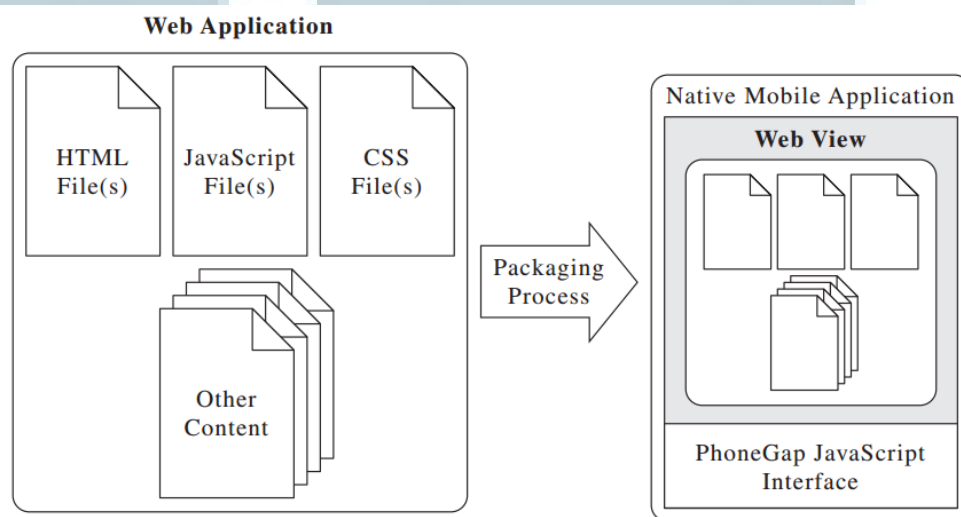


Gambar 2.2 Android Virtual Device

2.2 PhoneGap

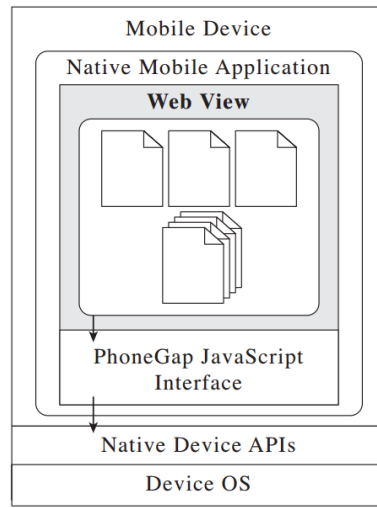
Menurut Wargo [4] PhoneGap merupakan suatu *framework open-source* yang berfungsi untuk membangun *cross-platform native applications* menggunakan bermacam-macam teknologi web seperti HTML5, CSS, dan Javascript. Tipe dari aplikasi ini disebut sebagai *hybrid application*.

Cara kerja PhoneGap adalah dengan cara mengubah *web application package* menjadi sebuah *native application*. Aplikasi yang telah dibuat akan ditampilkan dalam bentuk *web view* yang memungkinkan pengguna untuk melakukan interaksi dengan aplikasi tersebut.



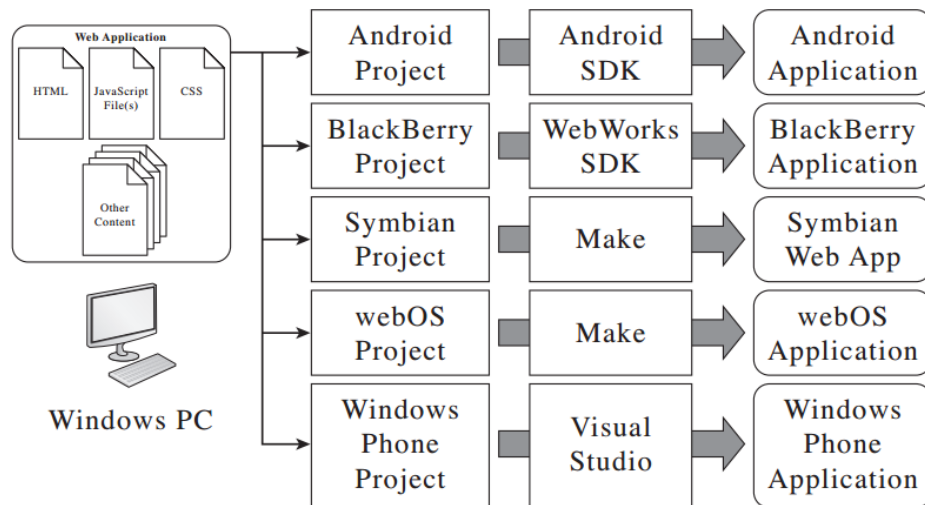
Gambar 2.3 Arsitektur Aplikasi PhoneGap

Tampilan *user interface* dari aplikasi yang dibuat oleh PhoneGap dibentuk oleh CSS yang diimplementasikan kedalam HTML5 dan untuk komunikasi data dapat menggunakan XML ataupun JSON.



Gambar 2.4 Interaksi *Device* pada Aplikasi PhoneGap

Pada Aplikasi yang telah dibangun dengan menggunakan PhoneGap, selanjutnya akan rubah dengan menggunakan SDK (*Software Development Kit*) berdasarkan target *platform* yang diinginkan agar aplikasi menjadi *native application*.



Gambar 2.5 Aplikasi PhoneGap diberbagai *platform*

Kini PhoneGap telah banyak mendukung banyak *platform* terkenal seperti Android, iOS, Windows Phone, BlackBerry, dan lainnya.

Setiap *platform* memiliki fungsi yang berbeda-beda, karena itulah PhoneGap menyediakan API (*Application Programming Interface*) untuk mempermudah pembangunan aplikasi, contohnya seperti, *accelerometer, camera, capture, compass, connection, contacts, device, events, file, geolocation, media, notification, dan storage*. API tersebut dirancang agar pengembang dapat mengakses fungsi *native application* melalui Javascript dengan *syntax* yang sama disemua *platform*, namun tidak semua *platform* dapat menggunakan API yang disediakan oleh PhoneGap karena keterbatasan *software* dan *hardware* di tiap *platform*.

Dalam penyimpanan data, PhoneGap menggunakan teknologi *local storage* yang terdapat pada HTML5.

2.3 Node.js

Merupakan suatu platform software yang dikembangkan oleh Ryan Lienhart Dahl. *Platform* ini digunakan untuk merancang aplikasi-aplikasi server side yang fleksibel di suatu jaringan. Node.js menggunakan bahasa pemrograman Javascript, dengan begitu Node.js dapat dengan mudah menghasilkan *throughput* melalui *non-blocking I/O*. Node.js memiliki fitur *built-in HTTP server library* yang membuat Node.js mampu menjadi sebuah web server tanpa bantuan software lainnya seperti Apache atau Nginx [5].

Semenjak pertama kali diluncurkan, Node.js telah menarik banyak perhatian. Salah satu alasannya adalah karena Node.js menggunakan bahasa pemrograman Javascript. Javascript merupakan salah satu bahasa pemrograman yang banyak digunakan [5].

2.4 **Websocket**

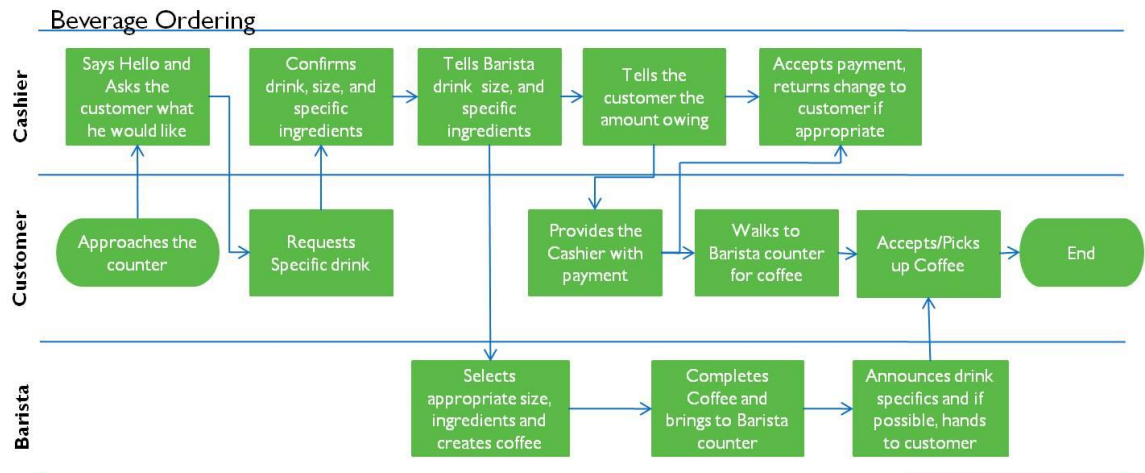
Websocket merupakan suatu teknologi *web* yang menyediakan komunikasi dari *client* ke *server* melalui sebuah koneksi *single* TCP tanpa harus membuat sebuah *request*. Dengan menggunakan *websocket*, *client* dan *server* dapat saling melakukan *push message* kapan saja [6].

2.5 **Coffee Shop (Kedai Kopi)**

Menurut wiseGEEK [7] *coffee shop* merupakan sebuah *café* kecil atau restoran kecil yang biasanya menjual kopi dan terkadang minuman non-alkohol, makanan kecil atau snack, dengan fasilitas yang menunjang di tempat tersebut.

Dari suatu pengamatan budaya, *coffee shop* banyak memberikan layanan sebagai pusat-pusat interaksi sosial: *coffee shop* dilihat memberi kesempatan kepada anggota-anggota sosial untuk berkumpul, berbicara, menulis, membaca, menghibur satu sama lain, atau membuang waktu, baik secara individu atau dalam kelompok kecil.

Berikut merupakan bisnis proses dari *Coffee Shop*:



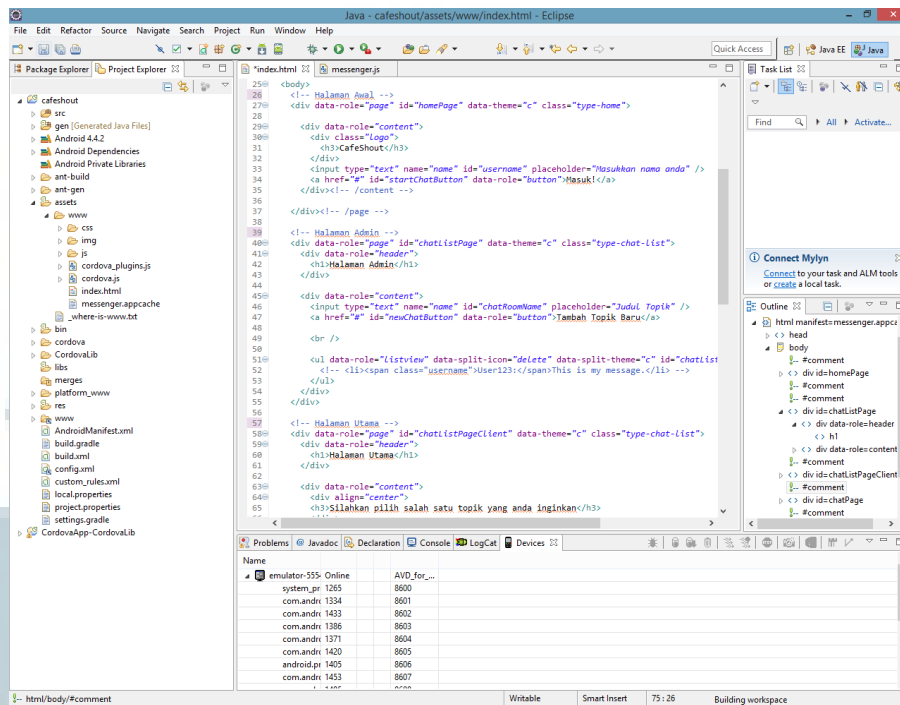
Gambar 2.6 Bisnis Proses *Coffee Shop*

(Sumber : <http://www.thetingleyadvantage.com>)

2.6 *Eclipse*

Eclipse merupakan suatu IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dijalankan di semua platform. Eclipse sendiri juga merupakan sebuah komunitas *open source*, yang memiliki proyek yang berfokus pada membangun sebuah platform pengembangan terbuka dari *extensible framework*, *tools*, dan *runtime* untuk membangun, menyebarkan, dan mengelola perangkat lunak di seluruh siklus hidup perangkat lunak tersebut.

Umumnya Eclipse digunakan untuk membuat sebuah program yang menggunakan bahasa pemrograman Java. Namun, Eclipse juga bisa digunakan untuk penggunaan bahasa pemrograman lainnya seperti C, C++, COBOL, Perl, PHP, Python, dan lain sebagainya.



Gambar 2.7 Eclipse IDE Luna

2.7 Java

Menurut Sun [8], Java merupakan sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi Sun, Java merupakan nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada computer *standalone* ataupun pada lingkungan jaringan. Untuk membuat program java dibutuhkan kompiler dan interpreter untuk program Java berbentuk JDK (*Java Development Kit*) yang diproduksi oleh Sun Microsystems.

2.8 Uji Validitas dan Reliabilitas

Validitas adalah suatu ukuran yang menunjukkan bahwa variabel yang diukur memang benar-benar variabel yang hendak diteliti oleh peneliti [9].

Sementara reliabilitas merupakan suatu ukuran yang menunjukkan bahwa alat ukur yang digunakan dalam penelitian keperilakuan mempunyai keandalan sebagai alat ukur, diantaranya diukur melalui konsistensi hasil pengukuran dari waktu ke waktu jika fenomena yang diukur tidak berubah [9].

Penelitian memerlukan data yang betul valid dan reliabel. Sebelum digunakan sebagai data penelitian primer, terlebih dahulu diujicobakan ke sampel uji coba penelitian. Uji coba ini dilakukan untuk memperoleh bukti sejauh mana ketepatan dan kecermatan alat ukur dalam melakukan fungsi ukurnya.

2.9 Uji Normalitas

Uji Normalitas bertujuan untuk mengetahui apakah masing-masing variabel berdistribusi normal atau tidak [10]. Uji normalitas diperlukan karena untuk melakukan pengujian-pengujian variabel lainnya dengan mengasumsikan bahwa nilai residual mengikuti distribusi normal. Jika asumsi ini dilanggar maka uji statistik menjadi tidak valid dan statistik parametrik tidak dapat digunakan.

Dari hasil uji normalitas ini dapat menentukan dengan metode apa penelitian dapat dilakukan tes hipotesis. Apabila data berdistribusi normal maka akan dilakukan uji-t (*t-test*), namun jika data tidak berdistribusi normal maka akan dilakukan uji Mann Whitney sebagai pengganti uji-t (*t-test*)

2.10 Uji Mann Whitney

Mann Whitney test merupakan uji non parametris yang digunakan untuk mengetahui perbedaan median 2 kelompok bebas apabila skala data variabel terikatnya adalah ordinal, interval, atau ratio tetapi tidak berdistribusi normal [11].

Mann Whitney test disebut juga dengan Wilcoxon Rank Sum Test. Merupakan pilihan uji non parametris apabila uji-t (*t-test*) tidak dapat dilakukan oleh karena asumsi normalitas tidak terpenuhi atau data tidak berdistribusi normal. Tetapi meskipun bentuk non parametris dari uji-t (*t-test*), uji *Mann Whitney* tidak menguji perbedaan *mean* (rata-rata) dua kelompok seperti layaknya uji-t (*t-test*), melainkan untuk menguji perbedaan *median* (nilai tengah) dua kelompok.

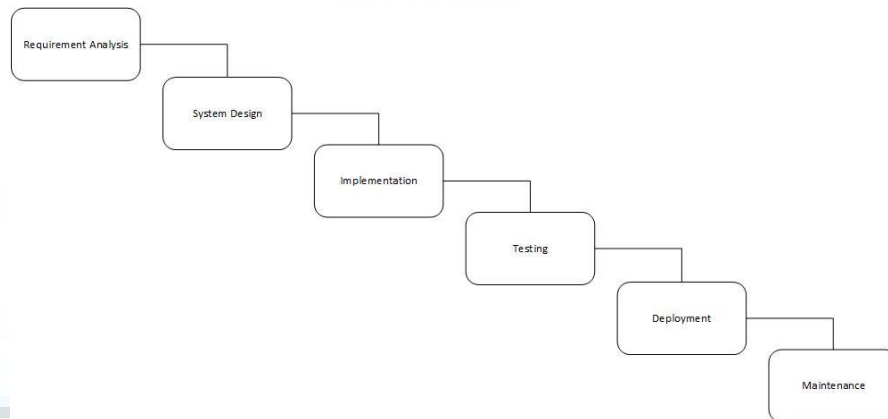
Tetapi beberapa ahli tetap menyatakan bahwasanya uji Mann Whitney tidak hanya menguji perbedaan median, melainkan juga menguji mean, karena dalam berbagai kasus, median kedua kelompok bisa saja sama, tetapi *P value* hasilnya kecil yaitu $< 0,05$ yang berarti ada perbedaan. Penyebabnya adalah karena mean kedua kelompok tersebut berbeda secara nyata. Maka dapat disimpulkan bahwa uji ini bukan hanya menguji perbedaan median, melainkan juga perbedaan mean.

2.11 SDLC (*Software Development Life Cycle*)

Menurut Roger S. Pressman [12], SDLC *Waterfall* atau metode yang memiliki nama lain “*Linear Sequential Model*” ini merupakan model yang paling banyak dipakai pada *Software Engineering* dan merupakan model yang muncul pertama kali pada tahun 1970. Model ini memberikan pendekatan-pendekatan sistematis dan berurutan bagi pengembangan piranti lunak. Pembangunan dengan metode *waterfall* memiliki tujuan yang berbeda-beda untuk setiap tahapan pembangunannya. Setelah tahapan pembangunan selesai, hasil pengembangan ke tahap berikutnya dan tidak akan kembali ke tahapan yang sebelumnya.

Keuntungan dari menggunakan metode ini adalah memungkinkan adanya pengelompokan kerja (*departementalization*) dan juga mengontrol manajerial. Setiap tahap pengembangan dapat diatur jadwalnya dan diberikan tenggat waktu, sehingga pembangunan akan lebih terstruktur dan selesai sesuai dengan rencana.

Sementara kerugian dari menggunakan metode ini adalah tidak memungkinkan adanya revisi. Setelah aplikasi sudah sampai pada tahap pengujian, sulit untuk kembali ke tahapan sebelumnya dan mengubah aplikasi, maka harus ada perencanaan yang matang pada saat pembangunan aplikasi dengan menggunakan metode ini.



Gambar 2.8 Model SDLC Waterfall

Berikut merupakan tahapan SDLC (*Software Development Life Cycle*) model *waterfall*:

1. *Requirement Analysis* (Analisis Kebutuhan)

Merupakan tahap analisis dan perencanaan dimana penulis mencari informasi mengenai permasalahan yang terkait dengan topik. Kebutuhan, kegunaan, dan batasan software juga harus didapatkan pada tahap ini. Informasi-informasi tersebut nantinya akan dianalisis untuk digunakan pada tahap selanjutnya.

2. *System Design* (Desain Sistem)

Merupakan tahap perancangan dimana penulis mencoba membuat solusi dari permasalahan yang diperoleh dari tahap analisis. Tahap ini membantu dalam memberikan spesifikasi kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation* (Implementasi)

Merupakan tahap implementasi dimana penulis mengimplementasikan rancangan dari tahap-tahap sebelumnya dan melakukan uji coba. Tahap ini penulis melakukan pemrograman. Pada tahap ini pengembang juga melakukan pemeriksaan terhadap modul yang telah dibuat, dan menentukan apakah aplikasi yang dibangun sudah memenuhi fungsi yang diinginkan.

4. *Testing* (Pengujian)

Merupakan tahap dimana aplikasi di ujicoba dan memastikan bahwa aplikasi telah mampu menyelesaikan permasalahan yang telah didapat dari tahap *requirement analysis* dan mengecek apakah masih aplikasi masih terdapat kesalahan atau tidak.

5. *Deployment* (Penyebaran)

Merupakan tahap dimana aplikasi telah berhasil di ujicoba dan telah dipergunakan oleh banyak orang.

6. *Maintenance* (Pemeliharaan)

Merupakan tahap dimana pengguna telah menggunakan aplikasi dan melakukan perbaikan apabila terdapat masalah.

2.12 Penelitian Terdahulu

Penelitian yang dilakukan oleh Gischa Joelita Eka Putri [1] dengan judul “Fungsi *Coffee Shop* Bagi Masyarakat Surabaya” yang menjelaskan bahwa fungsi *coffee shop* kini tidak hanya menjadi sebuah tempat untuk

mendapatkan minuman dan makanan namun kini telah menjadi sebuah tempat dimana seseorang dapat menemukan kebahagiaan, kepuasan, dan harga diri.

Penelitian ini dilakukan untuk menganalisis efek dari sebuah *coffee shop* sebagai sebuah indikator yang dapat berefek pada keinginan konsumen untuk datang ke toko mereka, sehingga dapat membuat suatu budaya baru.

Penelitian dilakukan dengan menggunakan metode riset secara kualitatif. Proses pengumpulan data dilakukan dengan cara melakukan wawancara terhadap 2 (dua) partisipan yang merupakan seorang *supervisor* sekaligus pemilik dari *coffee shop* yang bernama “Folks Coffee and Tea”, serta 20 (dua puluh) partisipan yang merupakan konsumen dari *coffee shop* tersebut. Selain itu peneliti juga melakukan observasi dan menggunakan data yang telah ada.

Hasil dari penelitian tersebut dapat menyimpulkan bahwa keberadaan *coffee shop* di Surabaya mulai dilihat konsumennya tidak hanya sebagai tempat untuk mendapatkan kopi, namun *coffee shop* kini berfungsi sebagai sebuah media komunikasi dan tempat untuk mengekspresikan diri dengan berinteraksi dengan orang lain atau dengan kelompok lain.

Sementara penelitian yang dilakukan oleh Allan C. Jeong yang berjudul “*The Sequential Analysis of Group Interaction and Critical Thinking in Online Threaded Discussions*” menunjukkan bahwa dengan cara memberikkan suatu topik permasalahan dapat mengundang perdebatan yang dapat mengundang

interaksi [13]. Pada perdebatan tersebut para partisipan akan terbagi menjadi 2 kelompok atau lebih dan masing-masing partisipan di tiap kelompok akan memiliki persamaan dalam perdebatan tersebut.

Partisipan dari riset tersebut berjumlah 34 orang yang terdiri dari Magister Administrasi Bisnis yang berisikan 10 wanita dan 24 pria. Para partisipan dimasukkan kedalam sebuah kelompok yang sudah ditentukan dan diperintahkan untuk berdebat tentang masalah etika selama 4 minggu pada sebuah papan buletin. Para partisipan dibagi menjadi 4 kelompok, kelompok 1 dan 2 akan memperdebatkan masalah pengawasan terhadap karyawan dan kelompok 3 akan memperdebatkan masalah pengumpulan informasi personal sebagai target periklanan, sementara kelompok 4 akan memperdebatkan tentang masalah kepekerjaan dan pemecatan berdasarkan perilaku diluar konteks kerja. Semua topik tersebut dipilih berdasarkan survey yang menentukan masalah yang paling kontroversial pada para partisipan. Hasil dari survey tersebut lalu di kalkulasikan menggunakan DAT (*The Discussion Analytics Tools*), alat tersebut digunakan untuk menghitung frekuensi terjadinya peristiwa, kemungkinan perubahan di setiap peristiwa, dan mengukur signifikansi statistik.

Hasil dari penelitian tersebut menyatakan bahwa dengan memberikan topik kontroversial yang dapat diperdebatkan dapat mengundang interaksi antar tiap kelompok yang berdebat. Setiap anggota kelompok yang sama menjadi memiliki persamaan pada setiap argumennya.