**BAB II  
TINJAUAN PUSTAKA**

2. 1. **Penelitian Terdahulu**

Tristantyo dalam penelitiannya berpendapat bahwa sistem jaringan komunikasi yang ada di Departemen teknik Sistem Komputer Universitas Diponegoro selama ini terdapat beberapa kelemahan yaitu, pertama sistem jaringan komunikasi melalui sms berantai yang rawan dengan informasi terputus di tengah jalan karena banyak faktor, kedua sistem jaringan komunikasi melalui jejaring sosial yang kurang efektif karena jejaring sosial yang bersifat publik dan tidak menyentuh langsung kepada sasaran informasi[1].

Anjuliani dan Astuti dalam penelitiannya berpendapat bahwa teknologi perangkat *mobile* dapat dimanfaatkan untuk mengakses berita dan informasi terkini serta berinteraksi dengan orang lain untuk bertukar pikiran maupun berbagi informasi. Ada suatu kendala yang terjadi dalam mendapatkan informasi tersebut, yaitu pengaksesan informasi yang kurang praktis dikarenakan sistem hanya sebatas berbasis aplikasi *web*. Tampilan web ini akan jauh lebih nyaman jika diakses melalui *web browser* perangkat komputer atau *laptop*. Selain kendala dalam pengaksesan informasi akademik, masih ada sejumlah kendala yang dihadapi mahasiswa, diantaranya belum dapat berinteraksi dengan sesama teman maupun dosen dalam sebuah forum diskusi yang khusus[2].

Mujab dalam penelitiannya berpendapat bahwa mahasiswa yang memiliki banyak kegiatan terkadang sulit mendapatkan informasi akademik karena mobilitas yang tinggi, walaupun mahasiswa dapat mengakses informasi akademik melalui perangkat bergerak namun hal tersebut kurang efisien, sistem informasi akademik yang sudah ada dibuat dengan *user interface* untuk tampilan *desktop* dan tidak menyediakan *user interface* untuk aplikasi *mobile.* Sebuah aplikasi berbasis *mobile web* dapat digunakan untuk memudahkan mahasiswa untuk mendapatkan informasi akademik yang dibutuhkan[3].

* 1. **Sistem Informasi**

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Informasi adalah salah satu sarana untuk memperkenalkan suatu perusahaan atau organisasi, sangat erat hubungannya dengan perkembangan organisasi yang masih dalam tahap perkembangan. Dengan tidak adanya informasi, maka suatu organisasi tidak akan pernah dapat cepat berkembang seperti apa yang diinginkan.

Sistem informasi dapat berupa gabungan dari beberapa elemen teknologi berbasis komputer yang saling berinteraksi dan bekerjasama berdasarkan suatu prosedur kerja yang telah ditetapkan, dimana memproses dan mengolah data menjadi suatu bentuk informasi yang dapat digunakan dalam mendukung keputusan[4].

Sesungguhnya, yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer-Based Information Systems* atau CBIS). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting.

Sistem Informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan[5].

* 1. **Android**

Android adalah tumpukan perangkat lunak untuk perangkat *mobile* yang mencakup sistem operasi, *middleware* dan aplikasi kunci. Android, sebuah *platform* perangkat bergerak *open source* berbasis pada sistem operasi Linux.

Memiliki kerangka aplikasi, peningkatan grafis, web *browser* yang terintegrasi, basis data relasional, dukungan media, LibWebCore web *browser*, berbagai konektivitas dan lebih banyak lagi aplikasi. Android bergantung pada Linux versi 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, tumpukan jaringan, dan model *driver*. Arsitektur Android terdiri dari aplikasi, *Kernel* Linux, perpustakaan, kerangka aplikasi, Android *Runtime*. Semua aplikasi ditulis menggunakan bahasa pemrograman Java. *Platform* perangkat bergerak Android akan menjadi lebih aman dari iPhone Apple atau perangkat lain dalam jangka panjang[6].

Membangun kontribusi dari komunitas *open-source* Linux dan lebih dari 300 perangkat keras, perangkat lunak, dan mitra operator, Android telah dengan cepat menjadi sistem operasi perangkat bergerak yang tumbuh paling cepat[7].



## **Android Studio**

Android Studio adalah IDE resmi Android. Ini bertujuan dibangun untuk Android untuk mempercepat pengembangan dan membantu Anda membangun aplikasi berkualitas tinggi untuk setiap perangkat Android[8].

Beberapa fitur yang Android Studio tawarkan adalah sebagai berikut:

1. Sistem *build* yang fleksibel berbasis *Gradle*,
2. *Emulator* yang cepat dan kaya-fitur,
3. Lingkungan yang cocok digunakan untuk pengembangan semua perangkat Android,
4. *Instant Run* untuk melakukan perubahan untuk aplikasi yang sedang berjalan tanpa membangun APK baru,
5. Integrasi *Code template* dan GitHub untuk membantu membangun fitur aplikasi umum dan contoh kode impor,
6. Alat pengujian ekstensif dan kerangka kerja,
7. Menyediakan peralatan *Lint* untuk merekam kinerja, penggunaan, kompatibilitas versi, dan permasalahan lainnya,
8. Mendukung C++ dan NDK,
9. Mendukung Google *Cloud Platform* yang memudahkan untuk melakukan integrasi Google *Cloud Messaging* dan *App Engine*.

Berikut penjelasan fungsi dari bagian-bagian yang berada di perangkat lunak Android Studio:

1. *Toolbar* memungkinkan untuk melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
2. *Navigation Bar* membantu untuk menavigasi melalui proyek dan membuka file untuk mengedit. Ini memberikan pandangan yang lebih kompak dari struktur terlihat di jendela *Project*.
3. *Editor Window* adalah di mana kode dibuat dan dimodifikasi. Tergantung pada jenis file yang dibuka, editor dapat berubah. Misalnya, saat melihat file layout, editor menampilkan *Editor Layout*.
4. *Tool Window* memberikan akses ke tugas-tugas tertentu seperti manajemen proyek, pencarian, kontrol versi, dan banyak lagi.
5. *Status Bar* menampilkan status proyek dan IDE itu sendiri, serta peringatan atau pesan[9].

* 1. **PHP *(Hypertext Preprocessor)***

PHP difokuskan pada *server-side scripting*, jadi pengguna dapat melakukan apa yang bisa dilakukan CGI dengan menggunakan PHP seperti mengambil data inputan *form*, meng-*generate* konten halaman dinamis, mengirim dan menerima *cookies* dan masih banyak lagi. Kemampuan dan *support*-nya untuk basis data juga sangat dapat diandalkan.

Berikut ini diberikan gambaran tentang cara kerja bahasa pemrograman PHP:

1. *Client*/*user* mengirimkan file PHP (menggunakan browser) melalui *web server* (seperti Internet Explorer, Mozilla Firefox, Google Chrome, dan lainnya).
2. *Web server* mendapatkan *request* atau permintaan dari *user* lalu meneruskan ke *server* melalui jaringan internet.
3. *Web server* lalu meneruskan permintaan file PHP tersebut ke PHP *proccessor*. PHP *proccessor* dapat berupa modul (bagian dari *web-server*) atau terpisah (sebagai CGI/*Fast* – CGI).
4. Permintaan diproses oleh PHP dan diteruskan ke basis data (jika terdapat permintaan ke basis data), kemudian hasilnya dikirim kembali ke *web server*.
5. *Web server* memaket kembali hasil tersebut dengan menambahkan HTTP *header* dan dikirim kembali ke *browser* melalui jaringan internet.
6. Browser memproses paket HTTP dan menampilkannya kembali kepada user sebagai file HTML[10].
   1. **MySQL**

MySQL *database server* adalah RDBMS (*Relational Database Management System*) yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menuntut *resource* yang besar. MySQL adalah basis data yang paling populer diantara basis data yang lain.

MySQL adalah program basis data yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*.

Beberapa kelebihan dan kuntungan MySQL, diantaraya adalah:

1. MySQL merupakan sistem manajemen basis data yang *open source*, yaitu *software* ini bersifat *free* atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
2. MySQL mempunyai performa yang tinggi tapi simpel.
3. MySQL dapat diakses melalui *protocol* ODBC (*Open Database Connectivity*) buatan Microsoft. Ini menyebabkan MySQL dapat diakses oleh banyak *software*.
4. Semua *client* dapat mengakses *server* dalam satu waktu tanpa harus menunggu yang lain untuk mengakses basis data.
5. MySQL merupakan basis data yang mampu menyimpan data berkapasitas besar, sampai berukuran *Gigabyte*.
6. MySQL dapat berjalan di berbagai sistem operasi seperti Linux, Windows, Solaris, dan lain-lain.

Pada mulanya MySQL bekerja pada *platform* Unix dan Linux. Namun, dengan perkembangannya, sekarang banyak bermunculan beberapa distro yang mampu berjalan pada beberapa *platform* yang bersifat *shareware* dan *corporate*.

MySQL sangat populer utamanya untuk aplikasi *web*. MySQL merupakan komponen basis data “M” yang ada di LAMP, BAMP, MAMP, dan WAMP (Linux/BSD/Mac/Windows-Apache-MySQL-PHP/Perl/Python). Selain itu, digunakan di tool *bugtracking* seperti Bugzilla.

MySQL terbukti kestabilannya karena digunakan oleh banyak proyek besar, seperti MediaWIki, Flickr, Facebook, Google, Nokia, dan Youtube. Selain itu, proyek CMS *free* seperti wordpress, Drupal, Joomla semuanya menggunakan MySQL sebagai *database server*-nya[11].

2. 3. **Pengujian Kotak Hitam *(Blackbox Testing)***

Perangkat lunak atau sistem yang sedang diuji dilihat sebagai “kotak hitam”. Pengujian kotak hitam adalah pengujian berdasarkan spesifikasi kebutuhan dan tidak perlu pemeriksaan terhadap kode. Ini murni dilakukan berdasarkan pengujian dari sudut pandang pelanggan untuk mengetahui set *input* dan *output* yang sudah diprediksi. Pengujian kotak hitam dilakukan pada produk yang sudah selesai seutuhnya.

Pengujian kotak hitam memainkan penting dalam pengujian perangkat lunak, untuk membantu validasi fungsi pada sistem. Pengujian kotak hitam dilakukan berdasarkan kebutuhan, sehingga beberapa kebutuhan yang belum selesai atau tidak terprediksi dapat diketahui dengan mudah dan dapat kemudian dibetulkan. Pengujian kotak hitam dilakukan berdasarkan pandangan akhir *user*. Yang paling penting dari pengujian kotak hitam adalah dapat mengetahui *input* yang *valid* dan *invalid* dari sudut pandang pelanggan.

Keuntungan utama dari pengujian kotak hitam adalah bahwa penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu, tidak hanya bahasa pemrograman tetapi juga pengetahuan tentang implementasi. Dalam pengujian kotak hitam baik *programmer* maupun penguji tidak bergantung satu sama lain. Keuntungan lain adalah bahwa pengujian kotak hitam dilakukan dari sudut pandang pengguna. Keuntungan yang paling jelas terlihat dari pengujian kotak hitam adalah membantu untuk membongkar spesifikasi yang ambigu dan tidak konsisten[12].

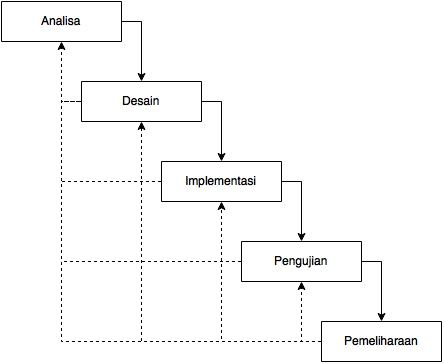
2. 8. **Metode *Waterfall***

Proses pembangunan perangkat lunak komputer dan sistem informasi selalu didikte oleh metodologi pengembangan yang berbeda. Sebuah metodologi pengembangan perangkat lunak mengacu pada kerangka kerja yang digunakan untuk perencanaan, pengelolaan, dan pengendalian proses pengembangan sistem informasi.

Secara formal, metodologi pengembangan perangkat lunak dikenal sebagai SDLC singkatan untuk *Software Development Life Cycle* dan digunakan dalam beberapa bidang industri seperti rekayasa sistem, rekayasa perangkat lunak, teknik mesin, ilmu komputer, ilmu komputasi dan, rekayasa terapan. Akibatnya, SDLC telah dipelajari dan diteliti oleh banyak peneliti dan praktisi di seluruh dunia, dan berbagai model telah dikemukakan, masing-masing dengan kelebihan dan kelemahan masing-masing.

Seperti model *waterfall*, salah satu model paling awal SDLC, terdiri dari lima fase yang berurutan yaitu: Analisa (*analysis*), desain (*design*), implementasi (*implementation*), pengujian (*testing*), dan pemeliharaan (*maintenance*).

Model *waterfall* SDLC adalah proses pengembangan perangkat lunak sekuensial di mana kemajuan dianggap sebagai aliran yang semakin ke bawah (mirip dengan air terjun) melalui daftar tahapan yang harus dijalankan agar berhasil membangun sebuah perangkat lunak komputer. Model *waterfall* mendefinisikan beberapa fase yang berurutan yang harus diselesaikan satu demi satu dan pindah ke tahap berikutnya hanya ketika fase sebelumnya yang benar-benar diselesaikan. Untuk alasan ini, model *waterfall* bersifat rekursif untuk setiap fase dapat terus menerus diulang sampai benar-benar sempurna. Kerangka model *waterfall* ditunjukkan pada gambar 2.1 berikut.



Gambar 2.1 Kerangka model *waterfall*

1. Tahap Analisis

Sering dikenal sebagai Persyaratan Spesifikasi Perangkat Lunak atau *Software Requirements Specification* (SRS) adalah deskripsi lengkap dan komprehensif dari perangkat lunak yang akan dikembangkan. Ini mengimplikasikan analis sistem dan tugas untuk menentukan persyaratan fungsional dan non-fungsional.

1. Tahap Desain

Tahap desain adalah proses perencanaan dan pemecahan masalah perangkat lunak. Tahap desain mengimplikasikan pengembang perangkat lunak dan desainer untuk menentukan rencana untuk sebuah solusi yang meliputi desain algoritma, desain arsitektur perangkat lunak, skema basis data konseptual dan desain diagram logis, desain konsep, desain antarmuka pengguna grafis, dan definisi struktur data.

1. Tahap Implementasi

Hal ini mengacu pada realisasi kebutuhan spesifikasi desain ke dalam *executable program*, basis data, *website*, atau komponen perangkat lunak melalui pemrograman dan *deployment*. Fase ini adalah di mana kode nyata ditulis dan disusun menjadi sebuah aplikasi operasional, dan di mana basis data dan file teks dibuat.

1. Tahap Pengujian

Hal ini juga dikenal sebagai verifikasi dan validasi yang merupakan proses untuk memeriksa bahwa perangkat lunak memenuhi persyaratan dan spesifikasi awal dan menyelesaikan tujuan yang telah ditetapkan. Verifikasi adalah proses evaluasi perangkat lunak untuk menentukan apakah produk dari tahap pengembangan yang diberikan memenuhi kondisi yang diinginkan pada awal fase; sementara, validasi adalah proses evaluasi perangkat lunak selama atau pada akhir proses pengembangan untuk menentukan apakah memenuhi persyaratan yang ditentukan. Selain itu, tahap pengujian adalah sarana untuk melakukan *debugging* di mana *bug* dan gangguan sistem ditemukan, dikoreksi, dan disempurnakan.

1. Tahap Pemeliharaan

Ini adalah proses memodifikasi perangkat lunak setelah di-deploy dan di-release untuk memperbaiki output, mengoreksi kesalahan, dan meningkatkan kinerja dan kualitas[13].



## **UML (*Unified Modeling Language*)**

*Unified Modeling Language* atau sering disingkat UML dapat didefinisikan sebagai bahasa standar untuk pemodelan umum dalam bidang rekayasa perangkat lunak berorientasi objek.

UML adalah alat untuk menentukan dan memvisualisasikan sistem perangkat lunak. Ini termasuk standar jenis diagram yang menggambarkan dan memetakan aplikasi komputer atau desain sistem basis data dan struktur. Penggunaan UML sebagai alat untuk menentukan struktur sistem adalah cara yang sangat berguna untuk mengelola sistem yang besar dan kompleks. Memiliki struktur yang terlihat jelas memudahkan untuk memperkenalkan kepada seseorang proyek yang sudah ada.

UML digunakan untuk menentukan, memvisualisasikan, memodifikasi, membangun dan mendokumentasikan artefak dalam pengembangan dari sistem perangkat lunak berorientasi objek. UML menawarkan cara standar untuk memvisualisasikan sistem *blueprints* arsitektur, termasuk unsur-unsur seperti:

1. Aktifitas
2. Aktor
3. Proses tugas
4. Skema basis data
5. Komponen (logis)
6. *Statement* bahasa pemrograman
7. Komponen perangkat lunak yang dapat digunakan kembali

UML menggabungkan teknik dan proses dari pemodelan data (*entity relationship diagram*), pemodelan tugas (*work flows*), pemodelan objek, dan pemodelan komponen. Hal ini dapat digunakan dengan semua proses, sepanjang siklus hidup pengembangan perangkat lunak, dan seluruh teknologi implementasi yang berbeda.

Diagram standar UML adalah: *Use case diagram*, *class diagram*, *sequence diagram*, *deployment diagram,* dan *activity diagram*.



### Use Case Diagram

Fungsionalitas yang disediakan oleh sistem basis data atau aplikasi komputer dapat diilustrasikan dengan sebuah *use case diagram*. Tujuan utamanya adalah untuk memvisualisasikan kebutuhan fungsional dari sistem, termasuk hubungan "aktor" (manusia yang akan berinteraksi dengan sistem) untuk proses penting, serta hubungan antara penggunaan case yang berbeda. Diagram *use case* biasanya digunakan untuk mengkomunikasikan fungsi tingkat tinggi dari sistem dan ruang lingkup sistem.

### Class Diagram

*Class diagram* menunjukkan bagaimana entitas yang berbeda (orang, benda, dan data) berhubungan satu sama lain. Hal ini digunakan untuk menampilkan *logical classes* dan *implementation classes*. *Class* adalah blok pengembangan utama pemodelan berorientasi objek. *Class diagram* adalah jenis diagram struktur statis di UML yang menggambarkan struktur dari suatu sistem dengan menunjukkan *class*, *atribute*, operasi (atau metode) dan hubungan antara kelas dari sistem.

### Sequence Diagram

*Sequence diagram* menunjukkan aliran rinci untuk penggunaan *case* tertentu atau bahkan hanya bagian dari penggunaan *case* tertentu. Ini menunjukkan panggilan antara objek yang berbeda dalam urutan mereka dan dapat ditunjukkan, pada tingkat rinci, panggilan yang berbeda untuk objek yang berbeda.

Sebuah *sequence diagram* memiliki dua dimensi: Dimensi vertikal menunjukkan urutan pesan / panggilan dalam urutan waktu terjadinya; dimensi horizontal menunjukkan objek pesan mana yang akan dikirim.

### Deployment Diagram

Diagram deployment menunjukkan bagaimana sistem akan secara fisik digunakan dalam lingkungan hardware. tujuannya adalah untuk menunjukkan di mana berbagai komponen dari sistem fisik akan berjalan dan bagaimana mereka akan berkomunikasi satu sama lain. Ini model penyebaran fisik artefak pada *node*.

Notasi dalam diagram *deployment* mencakup unsur-unsur notasi yang digunakan dalam diagram komponen, dengan beberapa penambahan, termasuk konsep *node*. Sebuah *node* mewakili salah satu mesin fisik atau *node* mesin virtual.

### Activity Diagram

Aliran prosedural kontrol antara dua atau lebih objek class saat memproses suatu kegiatan dapat ditampilkan dengan *activity diagram*. Hal ini dapat digunakan untuk memodelkan proses tugas tingkat tinggi di tingkat unit tugas, atau untuk memodelkan aksi internal *class* tingkat rendah.

*Activity diagram* adalah representasi grafis dari alur kerja kegiatan bertahap dan tindakan dengan dukungan untuk memilih iterasi dan konkurensi. *Activity diagram* digunakan untuk menggambarkan tugas dan alur kerja operasional dari komponen dalam suatu sistem. *Activity diagram* menunjukkan aliran kontrol[14].