

BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

Dalam bab ini akan diterapkan mengenai landasan teori yang bertujuan untuk membahas permasalahan yang diambil. Sesuai judul dalam pembuatan Tugas Akhir, yang akan dibuat penulis terdiri dari pengertian-pengertian sebagai berikut:

2.1.1 Pengertian Aplikasi

Pengertian aplikasi menurut Sutabri (2012:147), “Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimiliki.”

Pengertian aplikasi menurut Sujatmiko (2012:23), “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan untuk manusia dalam mengerjakan tugas-tugas tertentu, contohnya *Ms. Word, Ms. Excel.*”

Dari beberapa pengertian diatas dapat disimpulkan bahwa, Aplikasi adalah suatu perangkat lunak komputer yang difungsikan secara khusus untuk mengerjakan tugas yang diinginkan pengguna dengan tujuan mendapatkan hasil yang lebih akurat.

2.1.2 Pengertian Pengawasan dan Pengecekan (*Monitoring*)

Pengertian monitoring menurut Sutabri (2012), “monitoring juga didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan”.

Pengertian monitoring menurut Wrihatnolo (2008), “Monitoring dapat memberikan informasi keberlangsungan proses untuk menetapkan langkah menuju ke arah perbaikan yang berkesinambungan. Pada pelaksanaannya,



monitoring dilakukan ketika suatu proses sedang berlangsung. Level kajian sistem monitoring mengacu pada kegiatan per kegiatan dalam suatu bagian”.

Dari beberapa pengertian diatas dapat disimpulkan bahwa monitoring adalah proses pengumpulan dan analisis informasi kegiatan yang dilaksanakan. Sehingga dapat disesuaikan dengan apa yang telah ditetapkan.

2.1.3 Pengertian Kerusakan (Rusak)

Pengertian kerusakan menurut KBBI (Kamus Besar Bahasa Indonesia), “Arti kata rusak adalah sudah tidak sempurna lagi, atau tidak dapat berjalan lagi.

Dari pengertian diatas dapat disimpulkan bahwa rusak adalah kondisi yang sudah tidak sempurna lagi dan tidak dapat digunakan.

2.1.4 Pengertian Laboratorium Komputer

Dalam jurnal Hadyan (2014), “Laboratorium komputer merupakan sarana yang sangat dibutuhkan pada lingkungan perkantoran maupun pendidikan. Setiap kegiatan yang dilakukan pada laboratorium komputer harus mendapat pengawasan dari penanggung jawab laboratorium komputer itu sendiri, karena sangat penting untuk pendataan penggunaan laboratorium serta kapan saja laboratorium digunakan, laboratorium komputer juga membutuhkan suatu sistem informasi yang memudahkan proses kegiatan yang dilaksanakan pada laboratorium komputer, baik itu dalam pendidikan maupun perkantoran.”

Pengertian laboratorium komputer menurut KBBI (Kamus Besar Bahasa Indonesia), “Laboratorium adalah tempat atau kamar dan sebagainya tertentu yang dilengkapi dengan peralatan untuk mengadakan percobaan (penyidikan dan sebagainya).”

Dari beberapa pengertian diatas dapat disimpulkan bahwa laboratorium komputer adalah tempat yang sangat dibutuhkan yang dilengkapi dengan peralatan untuk mengembangkan keterampilan teknologi informasi.

2.1.4 Pengertian *Android*

Nazaruddin (2012:1), “*Android* merupakan sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. *Android* umum digunakan di *smartphone* dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di *Apple* dan BlackBerry OS.”

Hermawan (2011:1), “*Android* merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.”

Dari beberapa pengertian diatas dapat disimpulkan bahwa *android* adalah sistem operasi berbasis *linux* yang diperuntukan bagi pengguna perangkat bergerak seperti *smartphone* dan tablet PC.

2.2 Teori Khusus

Teori khusus adalah teori yang berkaitan dengan sejumlah fakta-fakta yang bersifat partikular. Sesuai judul dalam pembuatan Laporan Kerja Praktek, teori khusus yang ditulis adalah sebagai berikut:

2.2.1 Pengertian *Unified Modelling Language (UML)*

Menurut Pratama (2013 : 48), “*Unified Modelling Language(UML)* adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemrograman berorientasi objek”.

Menurut Shalahuddin et al. (2013 : 133), “*UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Jadi *Unified Modelling Language (UML)* adalah bahasa visual untuk pemodelan yang menjelaskan tentang analisis dan desain perangkat lunak dengan menggunakan diagram dan teks-teks pendukung.



Gambar 2.1 Logo *UML*

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

2.2.2 Jenis-Jenis Diagram *UML*

Adapun jenis-jenis diagram *UML* sebagai berikut:

1. Diagram Kelas, bersifat statis. Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.
2. Diagram Paket (*Package Diagram*), bersifat statis. Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen.
3. Diagram *Use Case*, bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas).



4. Diagram Interaksi dan *Sequence* (Urutan), bersifat dinamis. Dinamis urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*), bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* 1.4 yang menekankan organisasi struktural dari obyek-obyek yang menerima serta mengirim pesan.
6. Diagram *Statechart*, bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktivitas.
7. Diagram Aktivitas (*Activity Diagram*), bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.
8. Diagram Komponen (*Component Diagram*), bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem / perangkat lunak pada komponen-komponen yang telah ada sebelumnya.
9. Diagram *Deployment*(*Deployment Diagram*), bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*).

2.2.3 Pengertian *Use Case Diagram*

Menurut Pratama (2013:48), “*Use case diagram* menggambarkan aliran kegiatan dan proses bisnis yang dilakukan oleh pengguna (aktor)”.

Menurut Shalahuddin et al. (2013:155), “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

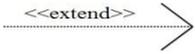
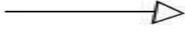
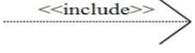
Jadi *use case* adalah pemodelan yang menggambarkan aliran kegiatan dari sistem informasi yang akan dibuat.

Komponen pembentuk diagram *use case* adalah:

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use case*, aktivitas/sarana yang disiapkan oleh bisnis / sistem.
3. Hubungan (*link*), actor mana saja yang terlibat dalam *use case* ini.

Adapun simbol-simbol menurut (Shalahuddin et al. (2013:155)) yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.1 Simbol *Use Case* Diagram

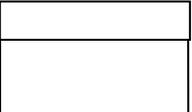
No.	Gambar	Nama	Keterangan
1		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
No	Gambar	Nama	Keterangan
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
3		<i>Association</i>	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4		<i>Extend</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.
5		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
6		<i>Include</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya sebagai syarat dijalankan use case.

2.2.4 Pengertian *Activity Diagram*

Menurut Rosa dan Shalahuddin (2013:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Adapun simbol-simbol menurut (Shalahuddin et al. (2013:161)) yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.2 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		Decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabung menjadi satu.
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6		Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

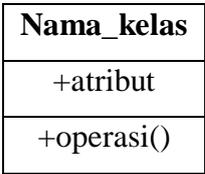
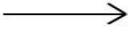
2.2.5 Pengertian *Class Diagram*

Menurut Pratama (2013:48), “*Class diagram* menggambarkan struktur kelas yang terdapat pada perangkat lunak yang akan dibangun”.

Menurut Shalahuddin et al. (2013:141), “*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Adapun simbol-simbol menurut (Shalahuddin et al. (2013:141)) yang digunakan dalam *class diagram* adalah sebagai berikut:

Tabel 2.3 Simbol *Class Diagram*

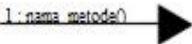
No.	Gambar	Nama	Keterangan
1		Kelas	Kelas pada struktur sistem.
2		Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3		Association	Relasi antarkelas dengan makna umum.
4		Directed Association	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
5		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6		Dependency	Relasi antarkelas dengan makna kebergantungan antarkelas
7		Aggregation	Relasi antarkelas dengan makna semua-bagian (whole-part).

2.2.6 Pengertian *Sequence Diagram*

Menurut Pratama (2013:48), “*Sequence diagram* menggambarkan *sequence* (aliran) pengiriman pesan (*message*) yang terjadi di aplikasi, sebagai bentuk interaksi dengan pengguna (*user*)”.

Adapun simbol-simbol menurut (Shalahuddin et al. (2013:162)) yang digunakan dalam *sequence diagram* adalah sebagai berikut:

Tabel 2.4 Simbol *Sequence Diagram*

No.	Gambar	Nama	Keterangan
1		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
2		Lifeline	Menyatakan kehidupan suatu objek.
3		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
4		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.

2.3 Refrensi Jurnal

Menurut penelitian Suryanti dan Welda (2017), Sistem monitoring penanganan kerusakan peralatan elektronik ini dibangun dengan tujuan dapat memantau penanganan kerusakan peralatan yang dilakukan oleh teknisi. Proses monitoring akan diawali dengan penginputan data kerusakan oleh bagian rumah tangga yang kemudian dapat langsung dilihat oleh bagian teknisi dan PK II. Jika perbaikan telah selesai dilakukan maka bagian teknisi akan menginputkan status dan tanggal selesainya perbaikan tersebut. Dengan adanya sistem ini diharapkan dapat meningkatkan kualitas pelayanan pada STMIK STIKOM Indonesia.

Menurut penelitian Al Hayubi, Arifi, Hasyim (2016), dalam jurnal yang berjudul Sistem Informasi Pengecekan dan Monitoring Laboratorium Komputer di STTNJ Berbasis Android dan Web. Proses pengecekan dan monitoring dirasa kurang maksimal karena belum memanfaatkan sistem informasi. Sehingga hal ini menyulitkan bagi asisten laboratorium karena asisten hanya berpegangan kepada list data pengecekan yang di tulis di kertas yang rentan hilang dan tercecer. Kesulitan ini juga dirasakan kepala laboratorium ketika meminta semua list pengecekan sebagai bahan monitoring. Maka dibutuhkannya sebuah sistem informasi yang dapat membantu memudahkan asisten laboratorium dalam hal pencatatan list pengecekan dan kepala laboratotium dapat memonitoring status laboratorium ready atau nonready secara realtime dan juga kepala laboratorium dapat mengetahui track record kinerja asisten laboratorium persemester.

Menurut penelitian Abdul, Ibnu Subroto, dan Kurniadi (2016), Masalah yang diangkat dalam paper ini adalah bagaimana merancang dan membangun sebuah sistem informasi yang digunakan untuk memudahkan pimpinan Dinas PU dalam memonitoring proyek dan mengefisiensikan proses kerja karyawan di setiap bidang unit kerja Dinas PU Sultra. Dengan kondisi tersebut, penulis memberikan sebuah solusi dengan mem-buatkan sebuah sistem informasi monitoring proyek untuk meningkatkan efisiensi kerja pada dinas pekerjaan umum provinsi Sulawesi Tenggara agar nantinya selain dapat memudahkan dalam hal melakukan pelaporan progres kerja juga akan mem-bantu sekretaris dinas PU dalam memantau pembangunan yang dilakukan di Sultra.

Menurut penelitian Bayu (2017), Merancang suatu program aplikasi tentunya membutuhkan suatu metode pengembangan perangkat lunak yang khusus untuk membantu selama proses pengerjaan. Secara keseluruhan teknologi informasi menjadikan metode pengembangan perangkat lunak bagian terpenting untuk pengembangan teknologi informasi. Metode pengembangan perangkat lunak pun menjadi semakin pesat perkembangannya. Sesuai dengan kebutuhan, perkembangan metode tersebut menjadi unsur penting dalam perkembangan tingkat kebutuhan teknologi informasi. Salah satu bentuk metode pengembangan perangkat lunak yang berkembang saat ini adalah metode agile. Dan salah satu



dari metode agile adalah *Extreme Programming* (XP). XP adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel. Tahapan-tahapan yang harus dilalui dengan XP yaitu, tahapan eksplorasi, perencanaan, iterasi pengembangan sistem, dan tahap produksi akhir. Pada penelitian ini pendekatan metodologi XP digunakan untuk melakukan perancangan pada sebuah aplikasi E-commerce berbasis M-commerce dalam penjualan buku. Metode ini mendukung pola pengembangan yang iteratif dan inkremental sehingga proses pengembangan dapat dilakukan lebih sesuai dengan kebutuhan dan dapat mengakomodasi adanya umpan balik maupun perubahan kebutuhan dari klien yang terkait dengan fungsi-fungsi pada aplikasi.

Menurut penelitian Wahyugi (2016), Penelitian ini bertujuan untuk menunjang layanan sistem informasi perpustakaan di berbagai universitas. Diantaranyadistribusi informasi seperti informasi tentang daftar buku, daftar judul tugas akhir. Info-info yang berkaitan tentang perpustakaan disampaikan melalui website, dimana info-info perpustakaan disimpan di dalam hosting website.