

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

1.1 Tinjauan Pustaka

Dengan pesatnya laju perkembangan teknologi, sistem penjadwalan pendadaran akan lebih efektif apabila menggunakan sebuah aplikasi khusus. Berdasarkan penelitian yang telah dilakukan sebelumnya, pembuatan aplikasi khusus ini akan membuat pengolahan data jadwal lebih efektif dan efisien.

Penelitian yang berjudul **“Sistem Penjadwalan Sidang Tugas Akhir Berbasis Web Dengan Pesan Pengingat Melalui SMS Dan Aplikasi pada Perangkat Android di Jurusan Teknik Elektro Universitas Diponegoro”**. Penelitian tersebut bertujuan ketika pengumuman sudah dikeluarkan mahasiswa tidak harus datang ke kampus untuk melihat hasil penjadwalan sidang tugas akhir. Sedangkan pembuatannya menggunakan Bahasa pemrograman *PHP*, basis data untuk jadwal maupun untuk *SMS Gateway* yang menggunakan *MySQL*, aplikasi *SMS Gateway* yang menggunakan *Gammu 1.25.0*. (Galoh Randicha, Wahyu Amien Syafei, Adian Fatchur Rochim, 2014)

Penelitian yang berjudul **“Otomatisasi Sistem Pendaftaran dan Penjadwalan Ujian Skripsi Fakultas Teknologi Informasi Universitas Kristen Satya Wacana”**. Tujuan dari penelitian tersebut untuk mempermudah pengolahan data sehingga lebih efisien. Hasil dari penelitian ini adalah sistem yang diimplementasikan dapat membantu pekerjaan dari *staff* yang memiliki tugas untuk jadwal ujian. Sedangkan dalam pembuatan menggunakan bahasa pemrograman *PHP*, *MySQL Server* sebagai databasenya dan aplikasi berbasis *Web*. (Thomas Adhi Nugroho, Christine Dewi, 2016)

Penelitian yang berjudul **“Pengembangan Sistem Informasi Penjadwalan dan Manajemen Keuangan Kegiatan Seminar dan Sidang Skripsi atau Tugas Akhir (Studi Kasus Program Studi Sistem Informasi UNIKOM)”**. Untuk meminimalisir masalah yang sering muncul dalam penjadwalan yang masih dilakukan secara manual. Dan mendapatkan data lebih akurat dalam proses perhitungan honorarium. Pengembangan aplikasi ini menggunakan Bahasa

pemrograman *PHP* dan *MySQL Server* sebagai databasenya. (Candra & Rajab, 2017)

Berdasarkan penelitian terlebih dahulu yang telah dijelaskan diatas, penulis dapat menyimpulkan beberapa perbedaan yang ada, yaitu:

1. Penjadwalan masih dilakukan secara manual oleh admin yang menambahkan data sesuai data yang ada dari mahasiswa.
2. Mahasiswa menentukan sendiri waktu pelaksanaan pendadaran.
3. Penguji masih ditentukan oleh mahasiswa ataupun program studi secara manual.
4. Tidak ada pembagian penguji sesuai kategori dari judul Tugas Akhir yang diambil mahasiswa.
5. Pembagian ruangan dan tanggal masih secara manual.

1.2 Landasan Teori

Untuk mendukung pembuatan laporan ini, maka perlu dikemukakan hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam membangun laporan ini.

2.2.1 Pengertian Penjadwalan

Menurut Kamus Besar Bahasa Indonesia, jadwal adalah pembagian waktu berdasarkan rencana atau urutan kerja, daftar atau table kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan pengertian penjadwalan itu sendiri adalah proses atau cara atau perbuatan menjadwalkan atau memasukkan kedalam jadwal.

2.2.2 Pengertian Pendadaran

Pendadaran adalah kegiatan tingkat akhir atau ujian yang ditujukan untuk mengevaluasi skripsi dan menguji pengetahuan apa yang kita pelajari selama perkuliahan. Ujian pendadaran dibagi menjadi dua : Ujian skripsi dan Ujian Komprehensif. Ujian skripsi mencakup seputar skripsi dan penelitian yang dilakukan. Ujian komprehensif yaitu ujian untuk mengetahui kemampuan secara

menyeluruh materi yang telah dipelajari selama perkuliahan. Fungsi pendadaran adalah untuk mengetes skripsi dan pengetahuan apa yang telah kita pelajari selama perkuliahan. [Sumber <https://eno-idone.blogspot.com/2016/05/ujian-pendadaran.html>]

2.2.3 Pengertian Tugas Akhir

Menurut Soedjono (1992: 12), tugas akhir merupakan suatu karya ilmiah berdasarkan suatu kegiatan penelitian mandiri mahasiswa, disusun dalam jangka waktu satu semester dibawah bimbingan seseorang dosen pembimbing dan dapat dibantu seorang pembantu pembimbing. Tugas akhir dilaksanakan secara mandiri oleh mahasiswa, dimaksudkan bahwa inisiatif perancangan penelitian, pelaksanaan penelitian dan penulisan laporan tugas akhir ada pada diri mahasiswa sendiri.

2.2.4 Pengertian Sistem

Menurut Djekky R. Djoht, sistem merupakan agregasi atau pengelompokan objek-objek yang dipersatukan oleh beberapa bentuk interaksi yang tetap atau saling tergantung, sekelompok unit yang berbeda, yang dikombinasikan sedemikian rupa oleh alam atau oleh seni sehingga membentuk suatu keseluruhan yang integral dan berfungsi, beroperasi, atau bergerak dalam satu kesatuan.

2.2.5 Pengertian Aplikasi

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas dapat disimpulkan bahwa aplikasi adalah berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. (Anisyah, 2000:3).

2.2.6 Pengertian Aplikasi Desktop

Desktop application atau aplikasi desktop adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi internet disuatu komputer otonom. (Dew Omenn, 2013)

Aplikasi berbasis Desktop merupakan aplikasi yang dijalankan pada masing-masing komputer atau klien. Aplikasi berbasis desktop harus diinstall terlebih dahulu ke dalam komputer agar dapat digunakan. Berdasarkan pengertian diatas, penulis menyimpulkan bahwa aplikasi berbasis desktop adalah aplikasi yang berjalan pada komputer yang dapat digunakan secara langsung ketika kode program selesai dikompilasi. (Rafyrpl101, 2013)

2.2.7 Metode System Development Life Cycle (SDLC)

SDLC (The System Development Life Cycle) merupakan suatu urutan proses yang bertahap dalam pengembangan suatu sistem. Pengembangan sistem yang dimaksud biasanya merujuk pada sistem komputer atau informasi. Siklus hidup pengembangan sistem merupakan langkah kerja yang berurutan dalam mengembangkan dan mengimplementasikan sistem informasi. SDLC merupakan suatu pola yang untuk mengembangkan sistem perangkat lunak yang terdiri dari tahapan rencana (planning), analisis (analysis), desain (design), implementasi (implementation), uji coba (testing), dan pengelolaan (maintenance) (Blanchard, B. S., & Fabrycky, W. J. (2006)).

2.2.8 Pengertian Unified Modeling Language (UML)

Untuk mendapatkan spesifikasi perangkat lunak yang sesuai dengan keinginan dan kebutuhan pengguna, para pengembang melakukan pemodelan-pemodelan secara visual. Langkah ini sering dinamakan pemodelan visual. Pemodelan visual adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi yang telah disepakati sebelumnya. Nugroho (2005:16-17).

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan,

membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*).

2.2.9 Pengertian *Entity Relationship Diagram (ERD)*

Menurut Brady dan Looman (2010), *Entity Relationship diagram (ERD)* “merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh Sistem Analisis dalam tahap analisis persyaratan proyek pengembangan sistem”.

Entity Relationship Diagram adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan *relationship* data (Sutanta, 2004).

Entity Relation Diagram (ERD) adalah notasi yang digunakan untuk melakukan aktivasi pemodelan data. Tujuan utama dari *Entity Relationship Diagram* (ERD) adalah mewakili objek data dan hubungan mereka (Fathansyah, 2007).

Komponen utama identifikasi untuk *Entity Relation Diagram* (ERD) berupa:

1. Entitas (*Entity*)

Entitas adalah suatu objek di dunia nyata yang dapat dibedakan dengan objek lainnya. Objek tersebut dapat berupa orang, benda ataupun hal lainnya.

2. Atribut (*Attribute*)

Atribut merupakan semua informasi yang berkaitan dengan entitas. Atribut sering dikenal dengan *property* dari suatu entitas atau objek.

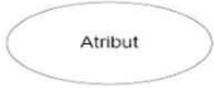
3. Relasi

Hubungan ditunjuk dengan garis yang diberi label yang menghubungkan objek. Sambungan antara objek dan hubungan dibangun dengan menggunakan kardinalitas dan modalitas. Relasi digambarkan dalam bentuk belah ketupat. Derajat kerdinalitas merupakan penjabaran dari hubungan antar entitas.

4. Garis

Adalah tanda garis yang digunakan untuk menghubungkan komponen-komponen *Entity Relation Diagram* (ERD) (Yanto,2016)

Notasi pada komponen utama identifikasi untuk *Entity Relation Diagram* (ERD) dapat dilihat pada gambar 2.1.

Notasi	Keterangan
	Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

Gambar 2. 1 Komponen utama ERD

2.2.10 Pengertian Validasi

Validasi merupakan proses penentuan apakah model konseptual simulasi benar-benar merupakan representasi akurat dari sistem nyata yang dimodelkan. Validasi model dapat pula dikatakan sebagai langkah dalam memvalidasi atau menguji apakah model yang telah disusun dapat merepresentasikan sistem nyata dengan benar. Suatu model dapat dikatakan valid ketika tidak memiliki perbedaan yang signifikan dengan sistem nyata yang diamati baik dari karakteristiknya maupun dari perilakunya. Validasi dapat dilakukan dengan menggunakan alat uji statistic yang meliputi uji keseragaman data output, uji kesamaan dua rata-rata, uji kesamaan dua variansi dan uji kecocokan distribusi (Law and Kelton,1991).

2.2.11 Bahasa Pemograman C#

C# merupakan salah satu bahasa pemograman yang diciptakan oleh Microsoft dan telah distandarisasi secara internasional oleh ECMA. Seperti halnya

bahasa pemrograman lain, C# dapat digunakan untuk membangun berbagai macam jenis aplikasi. Bahasa pemrograman C# dibuat berbasiskan bahasa C++ yang dipengaruhi oleh bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain dengan lebih sederhana (Abid Alfian Syakir, 2015).

2.2.12 Microsoft SQL Server

SQL Server adalah sebuah sistem manajemen *database* relasional yang memiliki kegunaan merancang sebuah aplikasi yang berhubungan dengan arsitektur *server* atau *client*. Pada umumnya *SQL Server* selalu dipergunakan di dunia bisnis dengan kelengkapan basis yang jauh lebih banyak namun memiliki skala kecil hingga skala menengah, akan tetapi sekarang ini lebih berkembang lagi sehingga menggunakan basis data dengan skala yang cukup besar. [Sumber <https://hendisaleh.wordpress.com/category/uncategorized/>]

2.2.13 Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *windows* ataupun aplikasi web.

Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic.NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual Fox Pro* dan *Visual SourceSafe*. *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan diatas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* untuk mengembangkan aplikasi *Silverlight*, aplikasi *Window Mobile* (yang berjalan diatas *.NET Compact Framework*). (Nolanda, Mutia, 2017)

2.2.14 Crystal Report

Crystal Report dirancang untuk membuat laporan yang dapat digunakan dalam bahasa pemrograman berbasis *windows*, seperti *Borland Delphi*, *Visual Basic*, *Visual C/C++* dan *Visual Interdev*. Menurut Hadi S, (2003) ada beberapa kelebihan dari *Crystal Report* ini adalah:

1. Dari segi pembuatan laporan, tidak terlalu rumit yang memungkinkan para programmer pemula sekalipun dapat membuat laporan yang sederhana tanpa melibatkan banyak kode pemrograman.
2. *Integrasi* dengan bahasa pemrograman lain yang memungkinkan dapat digunakan oleh banyak *programmer* dengan masing-masing keahlian.
3. Fasilitas impor hasil laporan yang mendukung format-format populer seperti *Microsoft Word*, *Microsoft Excell*, *Access*, *Adobe Acrobat Reader*, HTML dan sebagainya.

2.2.15 Black Box Testing

Dalam pengujian perangkat lunak ada dua yaitu white box testing dan black box testing. Dari kedua metode itu, pada skripsi dipilih menggunakan black box testing karena dianggap lebih tepat dibanding white box testing. Perangkat lunak memerlukan seperangkat tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini black box testing lebih sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Black box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineer untuk memperoleh input yang sepenuhnya akan melaksanakan persyaratan fungsional. (Roger S. Pressman, 2010). Untuk sebuah program. Black box testing berusaha untuk menemukan kesalahan dalam kategori berikut:

- a. Fungsi yang tidak benar atau fungsi yang hilang.
- b. Kesalahan antarmuka.
- c. Kesalahan dalam struktur data atau akses database eksterna.
- d. Kesalahan kinerja.
- e. Kesalahan inisialisasi dan pemutusan kesalahan