

## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Dalam pembahasan landasan teori ini penulis mengawali dengan menelaah dan mengenalisa penelitian – penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan penulis. Hal ini penulis lakukan untuk mendukung dan mempertajam penelitian dan penulisan laporan penelitian ini sendiri. Berikut ringkasan penelitian terdahulu yang mempunyai kaitan dengan penelitian penulis.

Penelitian yang dilakukan oleh Muhammad Hafidz Triyadi dan Bella Hardiana S.Kom., M.Kom dengan judul “Sistem Informasi *E-Learning* Dan Ujian Online Berbasis *Website* di SMP Negeri 5 Bandung” bertujuan merancang sistem informasi *e-learning* dan ujian *online* berbasis *website* di SMP Negeri 5 Bandung dimana aplikasi tersebut bisa membantu sistem belajar mengajar dan sistem ujian yang ada. Siswa dapat saling berbagi informasi dan dapat mengakses bahan-bahan belajar setiap saat dan berulang ulang. Selain itu, agar dapat memaksimalkan pekerjaan guru dalam proses pemberian materi/tugas, proses ujian siswa.

Persamaan penelitian ini dengan penelitian yang dilakukan oleh Muhammad Hafidz Triyadi et al adalah penelitian sama – sama melakukan analisis terhadap sistem informasi ujian yang sedang berjalan dan sama – sama mengangkat tema *Computer-based Test (CBT)*.

Perbedaan antara penelitian ini dengan penelitian yang dilakukan oleh Muhammad Hafidz Triyadi et al adalah perbedaan lingkup dalam proses

pembelajaran. Dalam penelitian ini lingkup proses pembelajarannya adalah hanya pada ujian. Sedangkan dalam penelitian yang dilakukan oleh Muhammad Hafidz Triyadi et al membahas keseluruhan proses pembelajaran siswa. [4]

Penelitian serupa juga dilakukan oleh Tono Hartono, S.Si, MT et al dengan judul “Aplikasi Simulasi *Try Out* Ujian Nasional Berbasis Android di SMA 1 Pasundan” penelitian ini dilakukan bertujuan untuk merancang dan membangun aplikasi simulasi *try out* ujian nasional yang merupakan aplikasi edukatif untuk membantu kegiatan belajar, khususnya dalam melatih kemampuan sebelum menghadapi Ujian Nasional. Pembangunan aplikasi simulasi *try out* ujian nasional dimulai dengan merancang UML, kemudian dilanjutkan dengan merancang tampilan lalu pengkodean program menggunakan *Eclipse* lalu dikompilasi dan diimplementasikan dengan menggunakan AVD 2.2. Aplikasi simulasi *try out* ujian nasional diharapkan dapat memberikan kemudahan bagi para siswa untuk mengetahui kemampuannya dan dapat meningkatkan ilmu pengetahuannya.

Persamaan dengan penelitian ini adalah tema yang diambil yaitu *Computer-based Test (CBT)*.

Perbedaan dengan penelitian ini adalah aplikasi yang dibangun dalam penelitian tersebut hanya untuk mengukur kemampuan siswa sebelum menghadapi Ujian Nasional yang artinya hasil dari ujian tidak akan digunakan untuk data akademik siswa. [5]

## **2.2 Teknologi Informasi**

Meliputi segala alat maupun metode yang terintegrasi untuk digunakan dalam menangkap dan menjaring data, menyimpan, mengolah, mengirim atau menyajikan

kebutuhan informasi secara elektronik kedalam berbagai format yang bermanfaat bagi pengguna. Teknologi ini dapat berupa kombinasi perangkat keras dan lunak dari komputer, non komputer maupun prosedur, operator dan para manajer dalam suatu sitem yang terpadu satu sama lain. [6]

### **2.3 Sistem Informasi**

Sistem informasi adalah sebuah sistem dalam sebuah organisasi yang merupakan kombinasi dari sekelompok orang, teknologi, prosedur – prosedur, fasilitas, media, dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi yang penting, memproses tipe transaksi rutin tertentu, memberi sinyal manajemen terhadap kejadian – kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang baik. [6]

### **2.4 *Computer Based Test***

*Computer Based Test* (CBT) adalah sistem evaluasi berbantuan komputer yang bertujuan untuk membantu guru dalam melaksanakan evaluasi, baik penskoran, pelaksanaan tes maupun efektifitas dan efisiensi pelaksanaannya. Tes nantinya akan berbantuan media dan pelaksanaannya pun menggunakan komputer.

Sistem *Computer Based Test* (CBT) atau pelaksanaan evaluasi dengan berbantuan komputer merupakan turunan atau pengembangan sistem computer assisted instructional (CAI) atau pembelajaran berbantuan komputer yang dikhususkan pada bidang garapan evaluasi meliputi kumpulan-kumpulan soal dan proses penskoran otomatis, media audio, video dan interaktif serta autorun. CBT merupakan sistem penilaian berdasarkan komputer serta bagian dari cakupan

Computer Assisted Instructional (CAI) yang dapat dilengkapi dengan audio, video, penskoran, dan layanan autorun. [3]

## **2.5 Learning Management System**

Merupakan salah satu bentuk pembelajaran elektronik yang digunakan untuk mendukung program belajar mengajar di banyak sekolah saat ini. Dapat dikatakan proses pembelajaran dengan penggunaan LMS adalah proses pembelajaran virtual. [7]

## **2.6 Sistem Informasi Akademik**

Sistem informasi akademik adalah suatu sistem yang dirancang untuk keperluan pengelolaan data - data akademik dengan penerapan teknologi komputer. [8] Selain mempermudah proses pengelolaan data, sistem informasi ini juga memperkecil biaya operasional yang harus dikeluarkan oleh lembaga terkait. [6]

## **2.7 Pendekatan Berorientasi Objek**

Pendekatan berorientasi objek adalah cara memandang persoalan menggunakan modelmodel yang diorganisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku suatu entitas. Pada pendekatan ini organisasi perangkat lunak adalah sebagai kumpulan objek diskrit yang saling bekerja sama, berkomunikasi dan berinteraksi menuju sasaran tertentu. [9]

## **2.8 Metode Pengembangan Sistem *Prototyping***

Metode *prototyping* adalah metode pengembangan sistem dimana *prototype* akan dibangun, diuji dan kemudian dibangun kembali seperlunya sampai mencapai hasil yang dapat diterima dari sistem atau produk yang lengkap. Metode ini baik digunakan dalam skenario di mana tidak semua persyaratan proyek diketahui secara

rinci sebelumnya. Metode ini adalah proses yang berulang, proses *trial-and-error* antara pengembang dan pengguna. [10]

## 2.9 Use case diagram

*Use case diagram* merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem. *Use case diagram* bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case diagram* juga bisa digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan bisa juga mempresentasikan sebuah interaksi aktor dengan sistem. Komponen tersebut kemudian menjelaskan komunikasi antara aktor, dengan sistem yang ada. Dengan demikian, *use case* dapat dipresentasikan dengan urutan yang sederhana, dan akan mudah dipahami oleh para konsumen. Manfaat dari *use case* sendiri adalah untuk memudahkan komunikasi dengan menggunakan *domain expert* dan juga *end user*, memberikan kepastian pemahaman yang pas tentang *requirement* atau juga kebutuhan sebuah sistem. [11]

## 2.10 Skenario use case

Skenario *Use Case* merupakan penjelasan secara tekstual dari sekumpulan skenario interaksi. Setiap skenario mendeskripsikan urutan aksi/langkah yang dilakukan aktor ketika berinteraksi dengan sistem, baik yang berhasil maupun gagal. [12]

## 2.11 Activity diagram

*Activity Diagram* merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. *Activity Diagram* juga digunakan untuk mendefinisikan atau mengelompokkan aluran tampilan dari sistem tersebut. *Activity*

*Diagram* memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Panah tersebut mengarah ke-urutan aktivitas yang terjadi dari awal hingga akhir. [13]

### **2.12 Class diagram**

*Class diagram* adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem class, atributnya, metode, dan hubungan antar objek. [14]

### **2.13 Sequence diagram**

*Sequence diagram* menjelaskan bagaimana suatu operasi itu dilakukan, pesan apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

*Sequence diagram* menampilkan interaksi antar objek dalam dua dimensi. Dimensi vertikal adalah poros waktu, dimana waktu berjalan ke arah bawah. Sedangkan dimensi horizontal merepresentasikan objek - objek individual. Tiap objek tersebut mempunyai waktu aktif yang direpresentasikan dengan kolom vertikal yang disebut dengan lifeline. Pesan direpresentasikan sebagai panah dari satu lifeline ke lifeline yang lain. [15]

### **2.14 Deployment Diagram**

*Deployment diagram* adalah diagram yang digunakan untuk memetakan perangkat lunak ke *processing node*. Menunjukkan konfigurasi elemen pemroses pada saat run time dan perangkat lunak yang ada di dalamnya. Diagram ini adalah salah satu diagram yang penting dalam implementasi perangkat lunak dan kadang-

kadang dibuat sebelum *coding*. Dengan menggunakan *deployment diagram*, dapat ditentukan ruang yang tersedia dan waktu eksekusi yang tersedia oleh perangkat keras. [16]

### **2.15 Black-box Testing**

Pada *Black-box Testing* dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Black-box Testing* ini lebih menguji ke Tampilan Luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh *Customer*. Pengujian ini tidak melihat dan menguji *source code* program. *Black-box Testing* bekerja dengan mengabaikan kontrol struktur sehingga perhatiannya hanya terfokus pada informasi *domain*. [17]

### **2.16 Entity Relationship Diagram**

*Entity Relationship Diagram* (ERD) merupakan sebuah model untuk menyusun database agar dapat menggambarkan data yang mempunyai relasi dengan database yang akan didesain. Diagram ER biasanya berhubungan langsung dengan diagram *data flow* untuk menampilkan konten *data store*. Ketiga hal tersebut dapat membantu memvisualisasikan bagaimana data saling terhubung dan berguna untuk mengonstruksi basis data relasional. [18]

### **2.17 Firewall**

*Firewall* adalah sistem keamanan jaringan komputer yang mampu melindungi dari serangan *virus*, *malware*, *spam*, dan serangan jenis yang lainnya. Dapat dikatakan juga bahwa, *firewall* merupakan perangkat lunak untuk mencegah akses yang dianggap ilegal atau tidak sah dari jaringan pribadi (*private network*).

Sehingga, tugas utama dari adanya *firewall* sendiri adalah untuk melakukan monitoring dan mengontrol semua akses masuk atau keluar koneksi jaringan berdasarkan aturan keamanan yang telah ditetapkan. [19]

## 2.18 Structured Query Language

SQL atau *Standard Query Language* adalah bahasa pemrograman yang digunakan dalam mengakses, mengubah, dan memanipulasi data yang berbasis *relasional*.

Tentu saja bahasa komputer dalam data basis *relasional* ini berdasarkan standar yang dikeluarkan oleh *American National Standard Institute* (ANSI). Standardisasi SQL sudah ada sejak tahun 1986 dan memang diinisiasi oleh ANSI. [20]

## 2.19 MySQL

MySQL adalah sebuah DBMS (*Database Management System*) menggunakan perintah SQL (*Structured Query Language*) yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis website. MySQL dibagi menjadi dua lisensi, pertama adalah *Free Software* dimana perangkat lunak dapat diakses oleh siapa saja. Dan kedua adalah *Shareware* dimana perangkat lunak berpemilik memiliki batasan dalam penggunaannya.

MySQL termasuk ke dalam RDBMS (*Relational Database Management System*). Sehingga, menggunakan tabel, kolom, baris, di dalam struktur database - nya. Jadi, dalam proses pengambilan data menggunakan metode *relational* database. Dan juga menjadi penghubung antara perangkat lunak dan *database server*. [21]

## 2.20 PHP

PHP (Perl Hypertext Preprocessor) adalah bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis”. Dengan menggunakan program PHP, sebuah website akan lebih interaktif dan dinamis.

Adapun kelebihan-kelebihan dari PHP yaitu:

1. PHP merupakan sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. PHP dapat berjalan pada web server yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada apache yang bersifat open source.
3. Karena sifatnya yang open source, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan developer yang siap membantu pengembangannya.
4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
5. PHP dapat berjalan pada 3 operating sistem, yaitu: Linux, unix, dan windows, dan juga dapat dijalankan secara runtime pada suatu console.

[22]

## 2.21 Framework

Framework adalah kerangka kerja. Framework juga dapat diartikan sebagai kumpulan script (terutama class dan function) yang dapat membantu developer/programmer dalam menangani berbagai masalah-masalah dalam pemrograman, seperti koneksi ke database, pembanggilan variable, file, dan lainlain

sehingga pekerjaan developer lebih fokus dan lebih cepat dalam membangun aplikasi. Framework adalah komponen pemrograman yang siap digunakan ulang kapan saja sehingga programmer tidak harus membuat script yang sama untuk tugas yang sama.

Secara sederhana bisa dijelaskan bahwa framework adalah kumpulan fungsi (libraries) sehingga seorang programmer tidak perlu membuat fungsi-fungsi dari awal dan biasanya disebut kumpulan library. [23]

## **2.22 Laravel**

Laravel adalah sebuah Framework PHP yang bersifat opensource yang ditulis oleh Taylor Otwell dengan lisensi dibawah MIT License. Laravel dibuat untuk membantu para developer khususnya dalam membuat sebuah web dengan sintak yang sederhana, elegan, ekspresif dan menyenangkan.

Laravel adalah aplikasi web dengan sintak yang ekspresif dan elegan. Dengan Laravel, tugas-tugas umum developer dapat dikurangi pada sebagian besar proyek - proyek web seperti routing, session dan caching. Disamping itu, laravel berusaha menggabungkan pengalaman-pengalaman development dalam bahasa lain, seperti Ruby on Rails, ASP.NET, MVC dan Sinatra. [24]

## **2.23 Vue.Js**

Vue (cara pengucapannya /vju:/, seperti **view**) adalah sebuah kerangka kerja nan progresif untuk membangun antarmuka pengguna. Tidak seperti beberapa kerangka kerja monolitik yang lain, Vue dirancang dari dasar sekali agar dapat diadopsi secara bertahap. Pustaka intinya difokuskan pada layer tampilan saja, dan sangat mudah untuk diintegrasikan dengan pustaka yang lain atau dengan proyek

yang sudah ada. Di sisi lain, Vue sangat mampu memberikan dan mendukung Single Page Application yang canggih ketika dikombinasikan dengan perkakas modern dan dukungan pustaka. [25]

#### **2.24 *Application Programming Interface (API)***

API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. Jadi, API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform.

Perumpamaan yang bisa digunakan untuk menjelaskan API adalah seorang pelayan di restoran. Tugas pelayan tersebut adalah menghubungkan tamu restoran dengan juru masak. Tamu cukup memesan makanan sesuai daftar menu yang ada dan pelayan memberitahukannya ke juru masak. Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan. [26]