

BAB II

KAJIAN LITERATUR

Bab ini menjelaskan secara singkat mengenai teori-teori yang berhubungan dengan kebutuhan aplikasi, informasi terkait organisasi yang diteliti yaitu Sekolah Tinggi Teknologi Terpadu Nurul Fikri, pengembangan sistem dengan metode *Incremental Development*, teori mengenai desain sistem menggunakan UML, teori mengenai pengembangan sistem menggunakan framework Laravel dan teori pengujian sistem menggunakan *Black Box Testing*.

2.1 Tinjauan Pustaka

2.1.1 Sekolah Tinggi Teknologi Terpadu Nurul Fikri

Sekolah Tinggi Teknologi Terpadu Nurul Fikri atau popular disebut STT-NF adalah salah satu perguruan tinggi di daerah Depok yang memadukan keilmuan praktis di bidang teknologi informasi dengan pengembangan kepribadian islami, kompeten dan berkarakter.

Pada tahun 2012, STT-NF resmi berdiri berdasarkan SK Menteri Pendidikan dan Kebudayaan Nomor 269/E/O/2012. Salah satu tujuan berdirinya STT-NF adalah “Mencetak sarjana komputer berakhlak mulia, profesional, dan bersertifikasi IT”. Tujuan itu untuk memenuhi kebutuhan sarjana teknologi informasi ditengah masyarakat, dengan konsisten mengusung nilai-nilai profesionalisme dan keislaman.

STT-NF mempunyai struktur organisasi yang cukup banyak yaitu Pembina, Ketua, Wakil Ketua I Bidang Akademik, Wakil Ketua II Bagian Sarana dan Prasarana (BSP), Sumber Daya Manusia (SDM), dan Keuangan, serta Wakil Ketua III Bidang Penerimaan Mahasiswa Baru (PMB). Salah satu bagian yang akan penulis bahas di dalam penelitian ini adalah Bagian Sarana dan Prasarana (BSP).

Bagian Sarana dan Prasana adalah salah satu bagian yang harus ada dalam suatu organisasi atau perusahaan untuk menentukan kebijakan dalam hal memfasilitasi suatu alat dan bahan agar tercapainya suatu tujuan pekerjaan. Proses dalam memenuhi fasilitas tersebut terdapat SOP atau Standar Operasional Prosedur yang harus dipenuhi. Demi tercapainya pemenuhan fasilitas, tentu nya perlu proses yang tidak memakan banyak waktu. Namun di

STT-NF ini proses pengajuan dalam memenuhi fasilitas tersebut masih kurang efektif karena pengajuan dilakukan secara manual yaitu dengan lembar form pengajuan barang.

Pada saat pengajuan, lembar form pengajuan barang harus diisi dengan perihal, unit, wakil ketua (waket) dan satuan kerja (satker), keterangan barang, jumlah barang, tanda tangan waket, tanda tangan kepala unit beserta lampiran gambar barang. Jika sudah memenuhi, pengaju akan memberikan dokumen tersebut ke Bagian Sarana dan Prasarana untuk dilengkapi data lainnya yang akan diajukan ke bagian Keuangan. Pengajuan barang di STT-NF sering mengalami kendala pada bagian verifikasi kepala unit yang tidak selalu ada di ruangan sehingga membutuhkan waktu lebih lama pada bagian pemrosesannya.

Pemakaian lembar form pengajuan barang juga tidak efektif karena memakan dana yang cukup banyak untuk pencetakannya. Dan juga resiko terburuk apabila form tersebut hilang atau rusak. Untuk itu, dengan adanya aplikasi form pengajuan barang akan sangat membantu bagian sarana dan prasarana dalam mengelola dan membackup data agar lebih efisien dan terstruktur.

2.1.2 Incremental Development

Pengertian dari *Incremental Development* adalah model pengembangan sistem pada *software engineering* berdasarkan *requirment software* yang dipecah menjadi beberapa fungsi atau bagian sehingga model pengembangannya secara bertahap. Disisi lain, model *Incremental* merupakan perbaikan dari model *waterfall* (Riyanti, 2019).

Tahapan pada *Incremental Development*, yaitu:

a. Requirment

Proses tahapan awal yang dilakukan pada model *Incremental* yaitu penentuan atau analisis kebutuhan.

b. Specification

Proses spesifikasi dimana menggunakan analisis kebutuhan sebagai acuannya.

c. *Architecture Design*

Perancangan *software* yang terbuka agar diterapkan sistem pembangunan bagian pada tahapan selanjutnya.

d. *Code*

Tahapan pengkodean setelah melakukan proses desain selanjutnya.

e. *Test*

Tahap pengujian dalam model ini.



Gambar 3: Tahapan Incremental Development

Kelebihan dari model *Incremental* yaitu:

1. Model manajemennya sederhana
2. Pengguna tidak perlu menunggu sampai seluruh sistem dikirim untuk mengambil keuntungan dari sistem tersebut. Jika *increment* yang pertama sudah memenuhi persyaratan, perangkat lunak dapat segera digunakan.
3. Resiko untuk kegagalan proyek secara keseluruhan lebih rendah. Walaupun masalah masih dapat ditemukan pada beberapa *increment*. Karena layanan dengan prioritas tertinggi diserahkan pertama dan *increment* berikutnya diintegrasikan dengannya, sangatlah penting bahwa layanan sistem yang paling penting mengalami pengujian yang ketat. Ini berarti bahwa pengguna akan memiliki kemungkinan kecil untuk memenuhi kegagalan perangkat lunak pada *increment* sistem yang paling bawah.
4. Nilai penggunaan dapat ditentukan pada setiap *increment* sehingga fungsionalitas sistem disediakan lebih awal.
5. Memiliki resiko lebih rendah terhadap keseluruhan pengembangan sistem.
6. Prioritas tertinggi pada pelayanan sistem adalah yang paling diuji.

Kekurangan dari model *Incremental* adalah:

1. Kemungkinan tiap bagian tidak dapat diintegrasikan.
2. Dapat menjadi *build* dan *fix model*, karena kemampuannya untuk selalu mendapat perubahan selama proses rekayasa berlangsung.
3. Harus *open architecture*.
4. Kesulitan untuk memetakan kebutuhan pengguna ke dalam rencana spesifikasi masing-masing hasil *increment*.

2.1.3 Laravel

Laravel dirilis sejak tahun 2011 dan mengalami pertumbuhan yang cukup pesat. Di tahun 2015, Laravel adalah *framework* yang paling banyak mendapatkan *rating* terbaik di Github.

Laravel merupakan salah satu dari banyak *framework* PHP yang dapat digunakan secara gratis. Laravel dikembangkan pada tahun 2011 oleh programmer asal Amerika yaitu Taylor Otwell dan dirilis dibawah lisensi MIT untuk memaksimalkan penggunaan PHP (PHP Hypertext Protocol) dalam pengembangan sebuah *website* (Rohi Abdulloh, 2017). Pengembangan Laravel berbasis MVC (*Model, View, Controller*) sehingga berfokus pada bagian *end-user* untuk memudahkan dalam penulisan maupun tampilan, meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal, biaya pemeliharaan, meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan *sintaks* yang ekspresif, jelas dan juga menghemat waktu.

Fitur pada Laravel adalah sebagai berikut (Muchammad Dwi, 2018):

- a. *Bundles* yaitu fitur dengan sistem pengemasan modular dan terdapat beragam aplikasi.
- b. *Eloquent ORM* yaitu penerapan PHP lanjutan yang menyediakan metode untuk mengatasi masalah pada hubungan objek basis data.
- c. *Reverse Routing* yaitu mendefinisikan URL yang akan menuju ke rute dengan definisi atau simbolis tertentu.
- d. *Restfull Controllers* yaitu pilihan untuk membedakan logika dalam melayani HTTP GET terhadap permintaan POST.

- e. *Class Auto Loading* yaitu *class-class* PHP yang menyediakan *loading* secara otomatis dan mencegah *loading* yang tidak diperlukan pada *class* PHP.
- f. *View Composer* yaitu kode unit *logical* yang dapat dieksekusi ketika *view* sedang *loading*.
- g. *IoC Container* yaitu memungkinkan obyek baru dihasilkan dengan pembalikan *controller*.
- h. *Migration* yaitu sistem kontrol yang disediakan pada *database* agar mengetahui perubahan dan memperbarui aplikasi.
- i. *Unit Testing* yaitu salah satu bagian penting untuk mendeteksi dan mencegah regresi atau pengaruh antara dua atau lebih variabel.
- j. *Automatic Pagination* yaitu menyederhanakan tugas penerapan halaman dari manual menjadi otomatis.

Keunggulan Laravel adalah:

- a. Laravel memiliki banyak fitur yang tidak dimiliki *framework* lain.
- b. Laravel merupakan *framework* PHP yang ekspresif atau bahasa yang mudah dimengerti untuk *programmer* pemula dalam mempelajari sintaks nya.
- c. Laravel menyediakan tampilan *template layout* yang ringan dan terdapat fitur tambahan pada *template*. Pengembang dapat menambahkan berbagai macam CSS, gambar, dan teks dengan struktur kode yang lengkap.
- d. Laravel memiliki berbagai macam *library* yang tidak ada pada *framework* PHP populer lainnya. Salah satunya *library* otentikasi yang menyediakan Bcrypt hashing yang merupakan fitur tercanggih di bidang otentikasi.
- e. Salah satu bagian terpenting yang dapat memudahkan dalam pemisahan antara logika dan tampilan aplikasi. Selain itu, performa yang dihasilkan optimal, dokumentasi lengkap dan terdapat banyak aplikasi tambahan.
- f. Tersedia *Tool Artisan* yang dapat melakukan tugas repetisi dan memproses pemrograman kompleks yang sering terabaikan oleh *developer*.
- g. Pembagian modul secara *independen* atau secara sendirinya mengadopsi prinsip PHP *modern* dan menjaga fungsi penting dari aplikasi. Membuat aplikasi menjadi lebih informatif, modular, dan responsif.
- h. Pengecekan menggunakan *Unit Testing* dalam jumlah banyak tanpa harus merusak satu pun fungsi aplikasi dan menjadi lebih responsif.

Kekurangan Laravel adalah:

- a. Ukuran *file* yang dihasilkan cukup memakan banyak memori.
- b. Membutuhkan koneksi internet untuk proses instalasi dan mengunduh *library*.
- c. Tidak semua *hosting* cocok dengan Laravel.

2.1.4 Laragon

Laragon adalah *universal development environment* untuk PHP, Node.js, Python, Java, Go, Ruby, yang *portable*, terisolasi, cepat, ringan, dan mudah dipakai. Laragon bisa dipakai sebagai pengganti XAMPP (Takagi Fujimaru, 2018).

Laragon dapat mengakses *project* melalui *app.test* daripada menggunakan *localhost/app*. Sistem Laragon terisolasi dari sistem operasi yang sudah ada sehingga apapun yang dilakukan dengan Laragon tidak akan mempengaruhi komputer lokal.

Laragon sudah mengotomatiskan banyak konfigurasi yang kompleks. Pengguna bisa menambah versi baru dari PHP, Python, Ruby, Java, Go, Apache, Nginx, MySQL, PostgreSQL, MongoDB, atau yang lainnya tanpa kesulitan yang berarti dan tanpa merusak sistem yang sudah ada.

Laragon memiliki arsitektur modern yang cocok untuk dipakai mengembangkan aplikasi *web* modern dan juga penggunaan Apache & Nginx sekaligus.

2.1.5 UML (*Unified Modeling Language*)

UML (Unified Modeling Language) merupakan bahasa untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak, serta untuk pemodelan bisnis dan bukan sistem perangkat lunak lainnya (Craig Larman, 2004).

UML diciptakan oleh *Object Management Group* yang diawali dengan versi 1.0 pada Januari 1997. Tujuan dari UML ini agar memudahkan *developer* melakukan penelusuran dan mempelajari kode program. UML juga dapat

menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya.

Structural Diagram

- a. *Class Diagram*, terdiri dari *class*, *interface*, *association*, dan *collaboration*. Diagram ini menggambarkan objek - objek yang ada di sistem.
- b. *Object Diagram*, diagram ini menggambarkan hasil instansi dari *class diagram*. Diagram ini digunakan untuk membuat *prototype*.
- c. *Component Diagram*, diagram ini menggambarkan kumpulan komponen dan hubungan antar komponen. Komponen terdiri dari *class*, *interface*, atau *collaboration*.
- d. *Deployment Diagram*, diagram ini menggambarkan kumpulan node dan hubungan antar node. Node adalah entitas fisik dimana komponen di-deploy. Entitas fisik ini dapat berupa server atau perangkat keras lainnya.

2.1.6 Black Box Testing

Pengujian sistem merupakan tahapan paling penting untuk mengetahui fungsionalitas pada suatu aplikasi yang telah dibuat dengan melibatkan perangkat lunak dan perangkat keras lainnya.

Salah satu teknik pengujian sistem yaitu dengan *Black Box Testing*. Pengujian *Black Box* sering juga disebut pengujian fungsional. Tujuannya untuk mengetahui struktur internal kode atau program pada perangkat lunak yang dibuat. Dalam pengujian ini, penguji menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya (Peni Kurniawati, 2018).

Kelebihan *Black Box Testing* yaitu:

- a. Efisien untuk segmen kode besar atau tidak perlu melihat kode sumber lebih detail.
- b. Mendeteksi kesalahan pada tampilan desain atau *user interface* dari sebuah perangkat lunak.
- c. Penguji tidak harus seorang *programmer*.

Kelemahan *Black Box Testing* yaitu:

- a. Cakupan terbatas karena hanya sebagian kecil dari skenario pengujian yang dilakukan
- b. Pengujian tidak efisien karena keberuntungan tester dari pengetahuan tentang perangkat lunak internal

2.1.7 UAT (*User Acceptance Test*)

User Acceptance Test (UAT) atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa *software* yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian (*testing*) sudah bisa dianggap memenuhi kebutuhan dari pengguna (Ade, 2016).

Proses UAT didasarkan pada dokumen *requirement* yang disepakati bersama. Dokumen *requirement* adalah dokumen yang berisi lingkup pekerjaan *software* yang harus dikembangkan, dengan demikian maka dokumen ini semestinya menjadi acuan untuk pengujian.

Proses dalam UAT adalah pemeriksaan dan pengujian terhadap hasil pekerjaan. Diperiksa apakah item-item yang ada dalam dokumen *requirement* sudah ada dalam *software* yang diuji atau tidak. Diuji apakah semua item yang telah ada telah dapat memenuhi kebutuhan penggunanya.

STT - NF

2.2 Penelitian Terkait

Dalam penelitian ini yang mengacu pada studi pustaka sebelumnya, ada beberapa penelitian terkait sebagai pembanding dengan masalah yang penulis tentukan. Hal ini bertujuan untuk mengetahui posisi penelitian yang dilakukan seperti melengkapi fitur yang belum ada pada penelitian sebelumnya ataupun membuat fitur baru. Daftar penelitian terkait dapat dilihat pada tabel berikut:

Tabel 1: Penelitian Terkait

Login	Upload	Download	Cetak	Notifikasi
	Deni, 2015	Deni, 2015	Deni, 2015	
Alvian, 2016	Alvian, 2016			
Rizal, 2018			Rizal, 2018	
Zahra, 2020				

1. 2015 – Deni Mahdiana

Jurnal berjudul “Analisa dan Rancangan Sistem Informasi Pengadaan Barang dengan Metodologi Berorientasi Obyek: Studi Kasus PT. Liga Indonesia”. Sistem informasi ini dibuat bertujuan untuk meminimalisir kesalahan pada pencatatan data serta memudahkan dalam mengontrol data masukan dan keluaran dalam penyajian informasi secara cepat dan akurat. Kebutuhan sistem tersebut diantaranya memasukkan data dan mencetak laporan pengadaan barang, penerimaan barang dan laporan pembayaran. Perancangannya menggunakan sistem berorientasi obyek Unified Modeling Language.

2. 2016 – Alvian Burhanuddin

Penelitian berjudul “Sistem Informasi Pengadaan Barang dan Jasa Melalui Penyedia di Unit Layanan Pengadaan Universitas Islam Negeri Maulana Malik Ibrahim Malang”. Sistem informasi ini dibuat bertujuan untuk memudahkan proses Administrasi yang sebelumnya dilakukan secara manual. Proses pembangunan aplikasi menggunakan Agile Unified Process. Bahasa pemrogramannya menggunakan MySQL dan PHP.

3. 2018 – Rizal Andian Rudiarko

Penelitian berjudul “Sistem Informasi Rencana dan Pelaporan Anggaran Program Unit Kerja di Yayasan Pendidikan Al Irsyad Surakarta”. Sistem informasi ini dibuat untuk mengoptimalkan penggunaan sistem yang sudah ada sebelumnya. Proses pembangunan aplikasinya menggunakan metode Waterfall. Bahasa pemrograman yang digunakan Laravel dan PHP.



STT - NF