

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam landasan teori penulis mengawali dengan menganalisa penelitian – penelitian terdahulu yang berkaitan dengan penelitian yang sedang dilakukan oleh penulis. Hal ini dilakukan untuk mempertajam penelitian dan penulisan pelaporan penelitian ini. Berikut penelitian terdahulu yang memiliki kaitan dengan penelitian penulis.

Seperti Penelitian yang dilakukan oleh Julian Chandra Wibawa dan Nizar Rabbi Radliya dengan judul Pembangunan Sistem Informasi Pengelolaan Surat Pada Program Studi Sistem Informasi UNIKOM dengan tujuan membangun dan merancang sistem informasi pengelolaan surat pada prodi sistem informasi Universitas Komputer Indonesia, dimana aplikasi ini membantu pengelolaan surat pada instansi pendidikan di Universitas Komputer Indonesia pada prodi Sistem Informasi. Sekretariat dapat membuat sebuah surat yang diajukan oleh pemohon dengan cara memasukkan data pada form yang sudah disediakan dan surat akan secara otomatis terbuat.

Persamaan pada penelitian ini dengan penelitian oleh Julian Chandra Wibawa dan Nizar Rabbi Radliya adalah penelitian sama – sama melakukan analisis pada Sistem Informasi pengelolaan dokumen dan surat dan sama – sama mengangkat tema pembuatan surat secara otomatis.

Perbedaan pada penelitian ini dengan penelitian oleh Julian Chandra Wibawa dan Nizar Rabbi Radliya adalah tidak adanya *template* yang bisa digunakan secara dinamis, pada penelitian tersebut peneliti hanya membuat sebuah surat saja dan yang sudah ditentukan berbeda dengan penelitian penulis yang bisa membuat sebuah surat lain sesuai dengan kebutuhan pengguna tanpa harus melakukan pembaharuan pada sistem informasinya [3].

Penelitian serupa juga dilakukan oleh Hardini Novianti dan Ali Bardadi dengan judul perancangan sistem informasi pengelolaan dokumen (studi kasus: fakultas ilmu komputer Universitas Sriwijaya). Dimana aplikasi ini untuk membantu pengelolaan dokumen masuk dan keluar pada fakultas ilmu komputer di Universitas Sriwijaya. Pada perancangan yang dilakukan oleh Hardini Novianti dan Ali Bardadi dengan pembuatan Data Flow Diagram untuk menggambarkan entitas dan proses apa saja yang diperlukan pada perancangan tersebut, kemudian dilanjutkan pembuatan ERD atau (*Entity Relationship Diagram*) dengan mengacu pada DFD yang sudah di rancang sebelumnya. Tujuan pada perancangan aplikasi ini adalah untuk menjamin bahwa dokumen tersedia dan beredar di suatu instansi adalah dokumen terkini (*update*) dan telah disahkan sebagai dokumen yang berlaku.

Persamaan dengan penelitian ini adalah mengangkat tema yang sama dan tujuan yang sama yakni pembuatan dan pengelolaan dokumen agar dokumen tersedia dan beredar di suatu instansi adalah dokumen terkini (*update*) dan telah disahkan sebagai dokumen yang berlaku.

Perbedaan dengan penelitian ini adalah lingkup penelitiannya adalah surat atau dokumen ini hanya berada pada lingkungan universitas atau fakultas saja akan

tetapi pada penelitian ini hanya fokus pada dokumen yang dapat digunakan oleh karyawan dan manajemen PT Someah Kreatif Nusantara.

2.2 Teknologi Informasi

Teknologi informasi adalah teknologi yang digunakan untuk mengolah data, termasuk memproses, memperoleh, menyusun, menyimpan, memanipulasi data dengan berbagai cara untuk menghasilkan informasi yang berkualitas tinggi, yaitu informasi yang relevan, akurat, dan tepat waktu, untuk keperluan pribadi, bisnis, dan tata kelola, dan merupakan informasi strategis untuk pengambilan keputusan. Teknologi informasi telah memberikan dampak yang sangat besar pada semua aspek kehidupan masyarakat karena telah menjadi bagian dari kehidupan masyarakat. Dalam teknologi informasi seperti itu, kemudahan berbagi informasi atau mencari informasi sangat penting, yang terlihat dari semakin berkembangnya teknologi internet dan jaringan. Internet sekarang menjadi kebutuhan untuk berbagi atau mencari informasi [4].

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 [5].

2.4 Metode Pengembangan *Prototyping*

Prototyping adalah metode pengembangan perangkat lunak yang merupakan model fisik dari cara kerja sistem, berfungsi sebagai versi awal sistem. Dengan menggunakan metode *prototyping* ini, dihasilkan sistem *prototyping* yang berperan sebagai perantara antara pengembang dan pengguna sehingga dapat berinteraksi selama berlangsungnya kegiatan pengembangan sistem informasi. Agar proses *prototyping* ini berjalan dengan lancar, perlu untuk mendefinisikan aturan sejak awal, yaitu pengembang dan pengguna harus memahami bahwa prototipe dibangun untuk menentukan persyaratan awal. Prototipe akan dihilangkan atau ditambahkan bagian-bagiannya agar sesuai dengan perencanaan dan analisis yang dilakukan oleh pengembang hingga uji coba berjalan bersamaan dengan proses pengembangan. Pembuatan prototipe oleh pengembang sistem bertujuan untuk mengumpulkan informasi dari pengguna sehingga pengguna dapat berinteraksi dengan model prototipe yang dikembangkan, karena prototipe menggambarkan versi awal sistem untuk digunakan dalam kelanjutan yang lebih besar dari sistem yang sebenarnya [6].

2.5 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek adalah cara memandang persoalan menggunakan model - model 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 [7].

2.6 Use case Diagram

Diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [8].

2.7 Activity Diagram

Activity diagram merupakan sebuah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram digunakan untuk menganalisis behaviour dengan *use case* yang lebih kompleks dan menunjukkan interaksi interaksi diantara mereka satu sama lain. Activity diagram biasanya digunakan untuk menggambarkan aktivitas bisnis yang lebih kompleks, dimana digambarkan hubungan antara satu *use case* dengan *use case* yang lainnya [9].

2.8 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 [10].

2.9 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 [11].

2.10 Entity Relationship Diagram

ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada Entity berikutnya [12].

2.11 No Structured Query Language (NoSQL)

NoSQL adalah sistem manajemen data non-relasional yang tidak memerlukan skema tetap. Istilah ini sendiri berasal dari singkatan “non-SQL” atau “not only SQL.” Umumnya, *database* RDBMS konvensional menggunakan sintaks SQL (structured query language) untuk menyimpan, mengambil, dan mengelola data pada *database*.

Sedangkan sistem *database* NoSQL mencakup lebih banyak teknologi basis data yang dapat menyimpan data terstruktur, semi terstruktur dan tidak terstruktur. Tujuan utama penggunaan *database NoSQL* adalah penyimpanan data terdistribusi dengan persyaratan penyimpanan data yang sangat tinggi [13].

2.12 MongoDB

MongoDB adalah salah satu jenis *database* NoSQL berbasis dokumen dengan menggunakan format file berupa JSON (JavaScript Object Notation). Jika dikomparasikan dengan penggunaan *database* SQL, dimana setiap data tersimpan dalam bentuk tabel. Sedangkan pada MongoDB, data akan disimpan ke dalam sebuah dokumen berformat JSON [14].

2.13 Javascript

JavaScript adalah bahasa untuk membuat program yang membuat dokumen HTML yang ditampilkan di browser lebih interaktif, tidak hanya estetis. JavaScript menyediakan beberapa fungsionalitas ke halaman web, sehingga dapat menjadi program yang melayani antarmuka web.

JavaScript merupakan bahasa script, yaitu bahasa yang tidak memerlukan kompiler untuk dapat menjalankannya, tetapi cukup dengan Interpreter. Tidak perlu ada proses kompilasi terlebih dahulu agar program dapat dijalankan. Browser web Netscape Navigator dan Internet Explorer adalah salah satu contoh dari salah banyak interpreter, karena kedua browser ini telah dilengkapi dengan Interpreter JavaScript. Tetapi tidak semua browser web dapat menjadi interpreter javascript karena belum tentu browser tersebut dilengkapi dengan interpreter JavaScript [15].

2.14 React JS

React JS adalah pustaka javascript yang dibuat oleh facebook, bukan sebuah *framework* MVC. React JS adalah bersifat composable user interface, yang berarti bahwa pengguna dapat membuat berbagai UI yang kemudian dapat dipecah menjadi beberapa komponen, dan juga komponen dapat digunakan kembali, yang

berarti bahwa komponen dapat digunakan kembali tanpa membangun kembali dari awal, React JS dapat digunakan untuk membangun Antarmuka pengguna untuk tampilan aplikasi satu halaman yang dirancang untuk menangani tampilan aplikasi web dan seluler [16].

2.15 Nest JS

Nest (NestJS) adalah kerangka kerja untuk membangun aplikasi sisi server Node.js yang efisien dan skalabel. Ini menggunakan JavaScript progresif, dibangun dengan dan sepenuhnya mendukung TypeScript (namun masih memungkinkan pengembang untuk membuat kode dalam JavaScript murni) dan menggabungkan elemen OOP (Pemrograman Berorientasi Objek), FP (Pemrograman Fungsional), dan FRP (Pemrograman Reaktif Fungsional) [17].

2.16 *Application Programming Interface (API)*

Application programming interface (API) merupakan suatu dokumentasi yang terdiri dari interface, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan programmer untuk “membongkar” suatu software, kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan sistem function. Proses ini dikelola melalui sistem operasi. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi [18].