**BAB II**

**TINJAUAN PUSTAKA**

**2.1 Penelitian Terkait**

Penelitian terkait terdiri dari ringkasan jurnal yang pernah melakukan penelitian dengan topik yang sama. Hasil dari penelitian sebelumnya ditunjukan pada tabel 2.1

**Tabel 2.1 Penelitian Terkait**

|  |  |  |
| --- | --- | --- |
| **No** | **Judul Penelitian** | **Garis Besar Penelitian** |
| 1 | Sistem Informasi *Monitoring* Perkuliahan Berbasis *Web* Di STMIK Sinar Nusantara Surakarta  (Wenda Priyanto, Didik Nugroho, Bebas Widada) | Membahas tentang sistem yang akan memonitoring pelaksanaan perkuliahan agar lebih terkontrol dan *up to date* [8]. |
| 2 | Analisis Sistem Informasi Manajemen Perkuliahan Pada Fakultas Ilmu Komputer Universitas Sriwijaya  (Mgs.Afriyan Firdaus, Firdaus, Ali Bardadali) | Menganalisa suatu sistem yang akan mempermudah top-level management dalam usaha manajemen dan evaluasi keseluruhan proses kuliah yang telah terjadi [9]. |
| 3 | Pembuatan Aplikasi Presensi Perkuliahan berbasis *fingerprint* Studi kasus : Jurusan Sistem informasi Institut Teknologi Sepuluh Nopember Surabaya  (Noval A.M, Febriliyan S, Radityo P.W) | Membahas mengenai sistem yang akan menanggulangi masalah kecurangan presensi yang sering terjadi serta memberikan kemudahan untuk melakukan rekapitulasi presensi mahasiswa selama proses belajar mengajar setiap harinya [10]. |
| 4 | Integrasi Sistem Presensi Finger Print Dan Sistem Sms Gateway Untuk *Monitoring* Kehadiran Siswa  (Dwi Agus Diartono) | Membahas mengenai sistem informasi yang dapat membaca atau menginput presensi dengan perangkat sidik jari siswa sekaligus dapat digunakan untuk *monitoring* kehadiran siswa bagi orang tua [11]. |
| 5 | Pencatatan dan Pemantauan Kehadiran Perkuliahan di Lingkungan Politeknik TelkomBerbasis RFID dan Aplikasi Web  (Tora Fahrudin) | Membahas mengenai model pencatatan kehadiran dengan RFID maupun aplikasi web dan aplikasi pemantauan kehadiran yang berupa *reporting realtime* [12] |

**2.2 Pengertian Sistem**

Sistem adalah seperangkat elemen yang membentuk kegiatan atau suatu prosedur atau tujuan-tujuan bersama dengan mengoperasikan data atau barang pada waktu rujukan waktu tertentu untuk menghasilkan informasi, energi dan barang [2].

**2.3 *Monitoring***

*Monitoring* adalah pemantauan yang dapat dijelaskan sebagai kesadaran tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu[3]. Kegiatan *monitoring* bisa diartikan sebagai suatu kegiatan memonitor atau mengawasi seluruh aktivitas yang dilakukan oleh seseorang [3]. Adapun beberapa tujuan dari sistem *monitoring* adalah [7] :

1. memastikan suatu proses dilakukan sesuai prosedur yang berlaku,
2. menyediakan probabilitas tinggi anakan keakuratan data bagi pelaku *monitoring*,
3. mengidentifikasi hasil yang tidak diinginkan pada suatu proses.

**2.4 Matrikulasi**

Matrikulasi berarti suatu proses yang membawa sebuah perguruan tinggi dan mahasiswa yang mendaftar untuk kredit perjanjian untuk tujuan mewujudkan tujuan pendidikan siswa. Perjanjian tersebut melibatkan tanggung jawab dari kedua pihak untuk mencapai tujuan tersebut melalui program-program yang didirikan di perguruan tinggi itu meliputi kebijakan, dan persyaratan- persyaratan [4].

**2.5 Metode Pengembangan Sistem**

Waterfall adalah serangkaian aktifitas proses-proses fundamental dari spesifikasi, development, validasi dan evolusi serta merepresentasikannya kedalam fase proses terpisah seperti kebutuhan spesifikasi, desain software, implementasi dan testing [5].



**Gambar 2.1. Model *Waterfall* Sommerville**

**(Sommerville, 2011, p.30)**

Tahapan dari model *waterfall* merefleksikan pokok-pokok dari aktifitas pengembangan [5]:

1. *Requirements analysis and definition*

Sistem layanan, kendala, dan tujuanditetapkan melalui konsultasi dengan pengguna sistem, kebutuhan tersebut kemudian ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

1. *System and software design*

Proses desain sistem mengalokasikan persyaratan baik untuk sistem perangkat keras atau perangkat lunak dengan mendirikan sebuah arsitektur sistem secara keseluruhan. Desain software melibatkan mengidentifikasi dan menggambarkan abstraksi sistem perangkat lunak yang mendasar.

1. *Implementation and unit testing*

Selama tahap ini, desain perangkat lunak disadari sebagai serangkaian program atau unit program. Unit testing memverifikasi bahwa setiap unit sesuai spesifikasi.

1. *Integration and system testing*

Unit program individu atau program diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah pengujian, sistem software diserahkan kepada pelanggan.

1. *Operation and Maintenance*

Ini adalah fase yang terpanjang. Sistem ini dipasang dan dimasukkan ke dalam penggunaan praktis. pemeliharaan melibatkan mengoreksi kesalahan yang tidak ditemukan dalam tahap awal siklus, meningkatkan implementasi unit sistem dan peningkatan sistem sebagai kebutuhan baru ditemukan.

**2.6 *Data Flow Diagram* (DFD)**

*Data Flow Diagram* (DFD) atau diagram alir data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil [6]. Salah satu keuntungan menggunakan diagram alir data adalah memudahkan pemakai atau user yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan [6]. Simbol-simbol DFD DFD hanya terdiri dari empat simbol. Simbol-simbol itu digunakan untuk (1) elemen-elemen lingkungan yang berhubungan dengan sistem, (2) proses, (3) arus data, serta (4) penyimpanan data.

Elemen-elemen lingkungan berada di luar batas sistem. Elemen-elemen ini menyediakan bagi sistem input data dan menerima output data sistem pada DFD, tidak dibuat perbedaan antara data dan informasi. Semua arus dipandang sebagai data.

1. Terminator

Nama terminator digunakan untuk menggambarkan elemen-elemen lingkungan, yang menandai titik-titik berkhirnya sistem. Termintor digambarkan dalam DFD dengan suatu kotak atau segi empat. Tiap simbol terminator diberi label nama elemen lingkungan.

**Gambar 2.2 *Terminator***

1. Proses

Proses adalah sesuatu yang mengubah input menjadi output. Proses dapat digambarkan dengan lingkaran, segi empat horizontal, atau segi empat tegak dengan sudut-sudut yang membulat. Tiap simbol proses diidentifikasikan dengan label. Teknik pembuatan label yang paling umum adalah dengan menggunakan kata kerja dan objek, tetapi dapat juga menggunakan nama sistem atau program komputer.

**Gambar 2.3 Proses**

3)Arus data

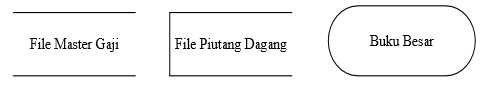
Arus data terdiri dari sekelompok elemen data yang berhubungan secara logis yang bergerak dari satu titik atau proses ke titik atau proses yang lain. Tanda panah digunakan untuk menggambarkan arus itu. Panah tersebut dapat digambar sebagai garis lurus atau garis lengkung.

Jumlah data yang diwakili oleh satu arus data dapat bervariasi dari satu elemen data tunggal hingga satu atau beberapa file. Contoh dari arus data yang terdiri dari satu elemen adalah jawaban atas pertanyaan manajer ke database untuk mendapatkan angka laba total bulan lalu. Arus data terdiri dari satu atau beberapa struktur data. Struktur adalah sekelompok elemen data yang menggambarkan suatu hal atau transaksi tertentu. Struktur paling mudah digambarkan sebagai pengaturan elemen-elemen data yang membentuk suatu catatan (record), atau sebagai sekelompok elemen-elemen yang berhubungan yang tercetak pada sebuah dokumen. Misalnya, baris jenis barang pada faktur adalah suatu struktur.

**Gambar 2.4 Arus Data**

1. Penyimpanan data

Jika data perlu dipertahankan karena suatu sebab, maka digunakan penyimpanan data. Dalam istilah DFD, penyimpanan data (data store) adalah suatu penampungan data. Bayangkan penyimpanan data sebagai data yang diam (data at rest). Pilihan untuk menggambarkan penyimpanan data sebagai satu set garis paralel, segi empat terbuka, atau bentuk lonjong.



**Gambar 2.5 Penyimpanan data**

Menggambar DFD sebenarnya hanyalah proses mengidentifikasi berbagai proses. Mengaitkannya dengan arus data untuk menunjukkan hubungan, mengidentifikasi terminator yang menyediakan input dan menerima output, serta menambahkan penyimpanan data jika perlu [7].

**2.7 ERD (*Entity Relationship Diagram*)**

Diagram hubungan entitas (*entity relationship diagram*) atau ERD, mendokumentasikan data perusahaan dengan mengidentifikasi jenis entitas dan hubungannya.

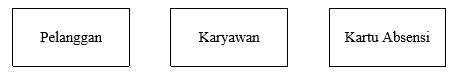
ERD disiapkan pada suatu titik dalam proses pengembangan sistem saat “gambaran besar” data ditentukan. Titik ini tiba:

* Saat eksekutif perusahaan terlibat dalam pembuatan model data untuk keseluruhan perusahaan dengan memperhatikan kebutuhan data untuk seluruh perusahaan.
* Saat eksekutif terlibat dalam pembuatan model untuk segmen operasi perusahaan yang besar, seperti area bisnis.
* Saat para spesialis informasi dan pemakai terlibat dalam pembuatan model data untuk area penerapan tertentu.

ERD karenanya merupakan pembuatan model data yang paling fleksibel, dapat diadaptasi untuk berbagai pendekatan yang mungkin diikuti perusahaan dalam pengembangan sistem [7].

**2.7.1 Jenis Entitas**

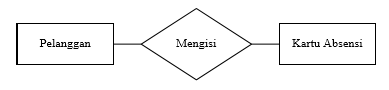
Jenis entitas (*entity type*) dapat berupa: (1) suatu elemen lingkungan, (2) sumber daya, atau (3) transaksi, yang begitu pentingnya bagi perusahaan sehingga didokumentasikan dengan data. Contoh jenis-jenis entitas adalah pelanggan, karyawan, dan kartu absensi. Jenis entitas didokumentasikan dalam ERD dengan segi empat, seperti tampak di bawah. Tiap segi empat diberi label nama dari jenis entitas.



**Gambar 2.6 Contoh jenis entitas**

**2.7.2 Hubungan**

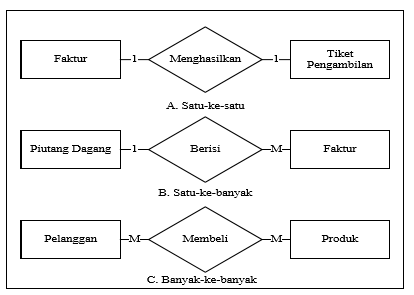
Hubungan (*relationship*) adalah suatu asosiasi yang ada antara dua jenis entitas. Hubungan digambarkan dengan bentuk belah-ketupat. Tiap belah-ketupat diberi label kata kerja. Pada contoh di bawah, seorang pegawai mengisi kartu absensi. Hubungan ini dapat juga dibaca mundur: kartu absensi diisi oleh pegawai.

****

**Gambar 2.7 Contoh hubungan entitas**

**2.7.3 Keterkaitan**

Banyaknya suatu entitas berhubungan dengan entitas lain disebut keterkaitan (*connectivity*). Ada tiga jenis keterkaitan - satu-ke-satu, satu ke-banyak, dan banyak-ke-banyak.



**Gambar 2.8 Contoh Keterkaitan Antar Entitas**

Suatu cara umum untuk menunjukkan keterkaitan adalah dengan menggunakan karakter 1 dan M, seperti tampak pada Gambar 2.8. Dalam Gambar 2.8 A, satu faktur menghasilkan satu tiket pengambilan, yang memungkinkan pekerja gudang mengambil barang yang tertera pada faktur (keterkaitan satu-ke-satu). Dalam Gambar 2.8 B, satu file piutang pelanggan berisi tagihan-tagihan untuk banyak faktur (keterkaitan satu-kebanyak), dan dalam Gambar 2.8 C banyak pelanggan membeli banyak produk (keterkaitan banyak-ke-banyak).

**2.7.4 Identifikasi dan Deskripsi Entitas**

Tiap kejadian dari tiap entitas perlu diidentifikasi dan dideskripsikan, dan ini dicapai dengan menggunakan atribut. Atribut adalah karakteristik dari suatu entitas. Misalnya, atribut seorang pelanggan meliputi nomor pelanggan, nama pelanggan, dan wilayah penjualan. Atribut-atribut tersebut sebenarnya adalah elemen-elemen data, dan masing-masing diberikan satu nilai tunggal, yang disebut nilai atribut.

Atribut yang mengidentifikasi entitas disebut identifiers. Contohnya adalah nomor pelanggan. Tidak ada dua pelanggan yang memiliki nomor yang sama. Atribut-atribut lain mendeskripsikan entitas, dan disebut descriptors. Contohnya adalah nama pelanggan dan wilayah penjualan. Identifiers sering tampak dalam ERD sebagai tulisan dengan garis bawah di dekat entitasnya [7].



**Gambar 2.9 Contoh Identifikasi dan Deskripsi Entitas**