

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Sistem Informasi

Sistem informasi adalah Suatu sistem yang dirancang untuk menyediakan informasi guna mendukung pengambilan keputusan pada kegiatan manajemen (perencanaan, penggerakan, pengorganisasian, dan pengendalian) dalam suatu organisasi [5]. Terdapat beberapa elemen kunci dalam sistem informasi yang efektif yaitu:

- A. Transformasi Data Menjadi Informasi. Data adalah fakta mentah yang belum diolah. Informasi adalah hasil pengolahan data yang memiliki makna dan nilai bagi penerimanya (manajemen) untuk digunakan dalam pengambilan keputusan. Tugas utama sistem informasi adalah mengolah data mentah (input) menjadi informasi yang berguna (output).
- B. Dukungan terhadap Fungsi Manajemen: Sistem informasi harus relevan dengan fungsi-fungsi manajemen, yaitu:
 - a) Perencanaan (*Planning*): Menyediakan data historis dan proyeksi untuk menyusun rencana masa depan.
 - b) Pengorganisasian (*Organizing*): Membantu alokasi sumber daya.
 - c) Penggerakan (*Actuating*): Membantu komunikasi dan koordinasi.
 - d) Pengawasan (*Controlling*): Memberikan umpan balik (*feedback*) mengenai kinerja organisasi dibandingkan dengan rencana.
- C. Integrasi Manusia dan Mesin: Sistem informasi adalah gabungan yang teratur antara sumber daya manusia, perlengkapan (teknologi atau komputer), dan prosedur-prosedur untuk menghimpun, menyimpan, dan mengolah data.

2.1.2 Manajemn Aset

Manajemen aset dipahami sebagai proses pengelolaan aset (milik individu, organisasi, atau perusahaan) yang dilakukan secara efektif dan efisien untuk mencapai tujuan tertentu. Putra menegaskan bahwa manajemen aset merupakan pengelolaan sekaligus siklus yang terjadi pada setiap perusahaan dan memerlukan sistem pengelolaan aset yang tepat. Sistem manajemen aset menurut para ahli adalah proses terstruktur untuk mengelola kekayaan organisasi secara efektif dan efisien, mencakup siklus perencanaan, akuisisi, penggunaan, pemeliharaan, hingga penghapusan aset untuk memaksimalkan nilai dan mendukung tujuan bisnis [5].

2.1.3 Website

Website adalah kumpulan halaman *web* yang saling terhubung dan dapat diakses melalui internet menggunakan sebuah alamat domain. *website* berfungsi sebagai media untuk menyampaikan informasi, menyediakan layanan, dan melakukan interaksi antara pengguna dan penyedia konten. Dengan berkembangnya teknologi internet, *website* telah menjadi sarana penting bagi individu dan organisasi untuk berkomunikasi, mempromosikan produk atau jasa, dan berinteraksi dengan audiens global [6].

2.1.4 Hypertext Preprocessor (*PHP*)

PHP (Hypertext Preprocessor) adalah bahasa pemrograman server-side yang dirancang khusus untuk pengembangan *web*. Menurut Lerdorf (2001), PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994 dan sejak itu telah berkembang menjadi salah satu bahasa pemrograman paling populer untuk *web* development. PHP memungkinkan pengembang untuk membuat halaman *web* dinamis

yang dapat berinteraksi dengan basis data dan menjalankan berbagai fungsi server-side lainnya [6].

2.1.5 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang berbasis pada SQL (*Structured Query Language*). MySQL dikembangkan oleh MySQL AB pada tahun 1995 dan sejak itu menjadi salah satu RDBMS paling populer di dunia. k adalah open-source, yang berarti dapat digunakan dan dimodifikasi secara gratis. Selain itu, MySQL juga tersedia dalam versi komersial dengan dukungan tambahan yang disediakan oleh Oracle Corporation, yang mengakuisisi MySQL AB pada tahun 2010 [6].

2.1.7 XAMPP

XAMPP adalah paket perangkat lunak yang mencakup Apache, MySQL, PHP, dan Perl, yang dirancang untuk memudahkan pengembangan *web* lokal di berbagai sistem operasi seperti Windows, macOS, dan Linux. XAMPP merupakan singkatan dari Cross-Platform (X), Apache, MySQL, PHP, dan Perl. Paket ini dikembangkan oleh Apache Friends dan tersedia sebagai perangkat lunak open-source, sehingga dapat diunduh dan digunakan secara gratis oleh pengembang *web* di seluruh dunia. XAMPP menyediakan lingkungan server yang lengkap di komputer lokal, memungkinkan pengembang untuk menguji dan mengembangkan aplikasi *web* sebelum dipublikasikan ke *server live* [6].

2.1.8. UML (Unified Modelling Language)

UML merupakan salah satu bahasa yang khas yang memiliki sintaks dan semantik. Komponen-komponen model kami harus mematuhi standar karena model-model yang kita buat berhubungan satu sama lain. Bahasa model digunakan oleh perancang sistem berorientasi objek untuk mendefinisikan, membuat, dan

mendokumentasikan sistem mereka [7]. Berikut ini adalah diagram yang ada pada UML:

a) *Use Case Diagram*

Use case diagram adalah ringkasan visual dari aktor sistem, banyak fungsi yang dibutuhkan oleh aktor-aktor ini, dan bagaimana fungsi-fungsi ini dapat berinteraksi satu sama lain. *Use case diagram* menggambarkan operasi utama sistem dan banyak pengguna yang berinteraksi dengannya[7]

b) *Activity Diagram*

Activity diagram adalah proses grafis yang digunakan untuk menunjukkan bagaimana aliran operasional atau bisnis komponen sistem. Terkadang activity diagram digunakan sebagai pengganti state machine diagram. Activity diagram bersifat independent dari class, alur aktivitas dalam use case, atau desain detail dari method[7]

c) *Component Diagram*

Component diagram menampilkan structural relationship antara component yang terdapat dalam suatu sistem. *Component* diagram digunakan pada sistem yang kompleks dan banyak memiliki component dimana antar component berkomunikasi melalui interface yang dihubungkan dengan menggunakan konektor. *View* dari *component* diagram biasa dilakukan dengan menggunakan dua pendekatan, yaitu *white box view* dan *black box view*. Pada *white box view* menampilkan struktur dari interface dan internal struktur. Sedangkan internal *black box view* digunakan provided interface dan required *interface*[7]

d) *Sequence Diagram*

Sequence diagram menekankan pada komunikasi pesan antara *lifeline* (objek) dan menggambarkan bagaimana dan dalam urutan apa item-item berinteraksi

satu sama lain. Sangat penting untuk diingat bahwa diagram urutan menggambarkan interaksi yang terjadi dalam keadaan tertentu. Representasi proses yang digunakan adalah garis-garis, sementara panah digunakan untuk menggambarkan interaksi[7]

e) *Class Diagram*

Class Diagram sering di sebut dengan diagram terstruktur, karena mampu menggambarkan apa yang seharusnya ada dalam sistem dengan berbagai komponen. Komponen tersebut antara lain *class* yang akan di program, objek utama, atau interaksi antara *class* dan objek.[7]

2.1.9. ERD (Entity Relationship Diagram)

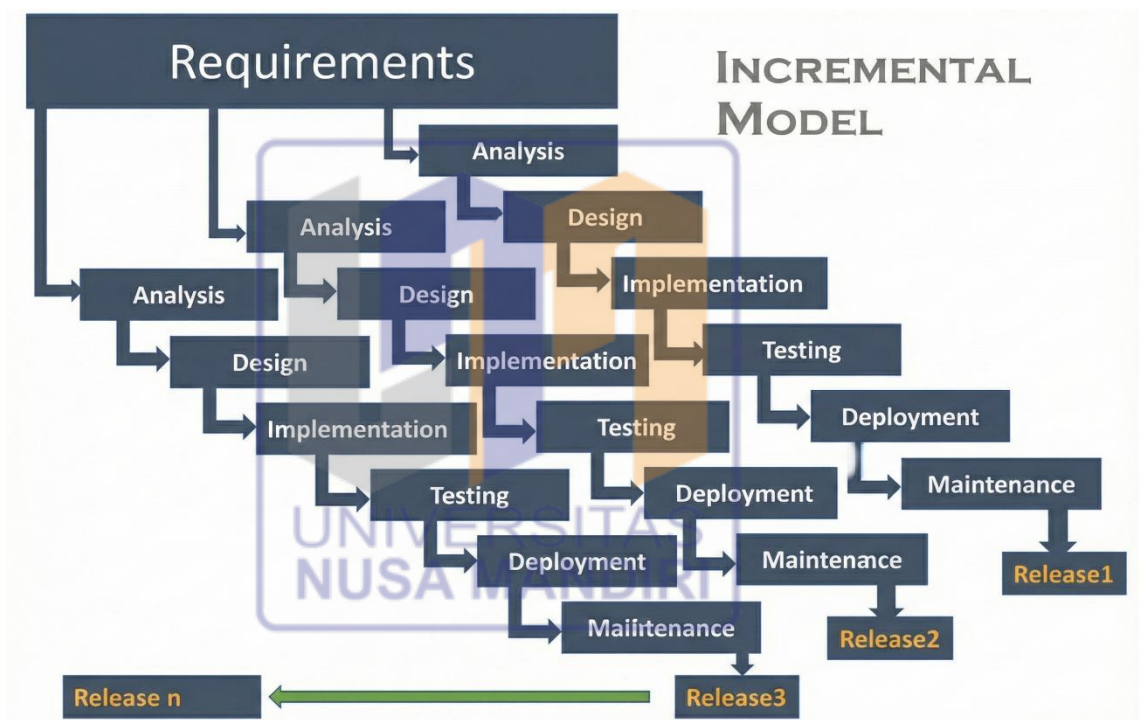
Sebuah model yang dikenal sebagai diagram hubungan entitas (ERD) menggunakan persepsi untuk merepresentasikan hubungan antara data dalam *database*. Isi *database* harus mematuhi aturan tertentu yang terkandung dalam model E-R. Pemetaan gaya memiliki aturan paling penting untuk menentukan berapa banyak entitas yang terkait dengan entitas lain melalui kumpulan relasi [8].

2.2.0. LRS (Logical Record Structure)

Model yang dapat dipahami digunakan untuk memperkenalkan LRS. Teknik pertama dimulai dengan hubungan antara dua model yang dapat ditransformasikan ke dalam LRS, sedangkan cara kedua dimulai dengan diagram ER dan dikonversi langsung ke dalam LRS. Nama dan tipe *record* berada di luar bidang tipe *record* yang ditentukan dalam LRS, berbeda dengan ERD. LRS terdiri dari hubungan tipe *record*. Hubungan ini mengarah dari satu tipe *record* ke tipe *record* lainnya. Bidang yang terlihat di kedua hubungan tipe *record* digunakan untuk mengidentifikasi banyak hubungan LRS [8].

2.2.1. Metode *Incremental*

Metode pengembangan *incremental* adalah pendekatan pengembangan perangkat lunak secara bertahap, di mana sistem dibangun dalam serangkaian peningkatan (*increment*) kecil yang fungsional, menggabungkan elemen model linier dan iteratif, sehingga menghasilkan produk yang dapat digunakan lebih cepat dan fleksibel dengan deteksi kesalahan dini, dengan fokus pada fitur utama dulu lalu fitur pendukungnya [9].



Sumber: [9]

Gambar II. 1 Model *Incremental*

Keterangan:

1. *Requirements* (Persyaratan Global)

Kotak besar di kiri atas menunjukkan bahwa kebutuhan atau persyaratan sistem secara keseluruhan biasanya didefinisikan di awal. Tim pengembang mengetahui gambaran besar apa yang ingin dibuat, namun pengerjaannya dibagi-bagi berdasarkan prioritas fitur.

2. Pemecahan Menjadi Beberapa Increment (*Release*)

Diagram menunjukkan adanya jalur-jalur paralel yang berujung pada *Release 1*, *Release 2*, *Release 3*, hingga *Release n*. Setiap jalur ini disebut sebagai increment.

- a) *Release 1* (Core Product): Biasanya berisi fitur paling dasar dan krusial. Begitu siklus ini selesai, pengguna sudah bisa menggunakan produk tersebut meskipun fiturnya belum lengkap.
- b) *Release 2 & 3*: Menambahkan fitur baru atau menyempurnakan fitur yang sudah ada dari *Release 1*.
- c) *Release n*: Adalah tahap akhir di mana sistem sudah lengkap sepenuhnya sesuai rencana awal.

3. Siklus Pengembangan (*Mini-Waterfall*)

Setiap increment (setiap anak tangga dalam diagram) melalui siklus pengembangan *software* (SDLC) yang lengkap dan standar, yaitu:

- a) *Analysis*: Menganalisis kebutuhan spesifik untuk bagian fitur tersebut.
- b) *Design*: Merancang teknis dan antarmuka untuk fitur tersebut.
- c) *Implementation*: Penulisan kode program (*coding*).
- d) *Testing*: Menguji apakah fitur tersebut berjalan lancar dan bebas bug.
- e) *Deployment*: Menerapkan atau merilis fitur tersebut ke pengguna.
- f) *Maintenance*: Pemeliharaan setelah fitur dirilis.

Siklus untuk *Release 2* dimulai sedikit setelah *Release 1* berjalan (*staggered*). Ini memungkinkan tim untuk bekerja secara paralel atau berkesinambungan. Ketika *Release 1* sedang dalam tahap *testing* atau *deployment*, tim mungkin sudah mulai melakukan *analysis* atau *design* untuk *Release 2*.

2.2 Penelitian Terkait

Model pengembangan *incremental* digunakan dalam pengembangan SIsKA. Tahap evaluasi dilakukan dengan teknik Focus Group Discussions untuk memvalidasi masukan-masukan pengguna untuk tahap pengembangannya. Evaluasi dilakukan terhadap 20 responden, yang dipilih dengan menggunakan teknik simple random sampling dari pengguna aktif SIsKA karyawan, integrasi sistem yang baik, dan ketersediaan infrastruktur teknologi yang memadai [10].

Target sasaran SMK-IT bidang keahlian Sistem Informasi Jaringan dan Aplikasi (SIJA) dan Rekayasa Perangkat Lunak (RPL). Berdasarkan data SMK yang ada di Kota Bogor maka perlu untuk dilakukan pengelompokan terhadap bidang kejuruan yang akan menjadi landasan target sasaran. Adapun pengelompokan terhadap bidang kejuruan disertai sebaran SMK menggunakan pendekatan teknologi SIG (Sistem Informasi Geografis). SIG yang akan diimplementasikan merupakan sistem pemetaan terhadap titik-titik koordinat SMK. Pembangunan SIG SMK menggunakan metode *incremental* model [11].

SMA Uswatun Hasanah ini merupakan salah satu sekolah umum dengan memberikan nilai lebih dalam hal keagamaan yang penerimaan peserta didik barunya masih dilakukan secara manual dengan membutuhkan jumlah panitia yang tidak sedikit, biaya yang mahal dan orang tua atau wali murid harus datang serta daftar langsung ke sekolah. Maka dari itu dibutuhkan suatu sistem informasi yang dapat membantu menyelesaikan permasalahan yang ada pada SMA Uswatun Hasanah [12].

Perbedaan kecepatan ini dipengaruhi beberapa faktor, salah satu faktornya adalah cara atau gaya belajar dari individu tersebut. Gaya belajar memiliki 3 jenis yaitu visual, audio dan kinestetik, pelajar dapat mengetahui gaya belajar yang sesuai dengan cara melakukan tes dengan menjawab pertanyaan-pertanyaan yang berkaitan dengan cara

belajar individu. Tujuan penelitian ini yaitu membuat aplikasi identifikasi gaya belajar berbasis *web* menggunakan metode *incremental* [13].

Penelitian ini mengimplementasikan Sistem Informasi Pendanaan Usaha Mikro Kecil (PUMK) menggunakan metode *incremental* dalam upaya meningkatkan efisiensi pengelolaan usaha mikro kecil (UMK) di PT. Angkasa Pura II Kantor Cabang Jambi. Sistem diimplementasikan berdasarkan komunikasi yang dengan pihak Angkasa Pura II Kantor Cabang Jambi, kemudian sistem diuji menggunakan black box testing dan User Acceptance Testing (UAT). Tahap pre- increment melibatkan identifikasi dan analisis kebutuhan sistem, yang kemudian dirancang sebagai basis pengembangan [14].

Pada umumnya suatu organisasi tidak melakukan sentralisasi data sehingga tidak ada acuan dalam pembagian informasi yang bisa dilihat oleh seluruh pihak yang membutuhkan data dan menjadi kendala atau masalah dikemudian hari salah satunya ada informasi palsu dan tidak akurat [15].

