

## BAB II LANDASAN TEORI

### 2.1 Landasan Teori

#### 2.1.1 Website

*Website* merupakan kumpulan dari halaman digital yang berisikan informasi berupa sebuah teks, animasi, gambar, suara dan video atau kombinasi dari semuanya yang terkoneksi oleh internet, sehingga dapat dilihat oleh siapapun yang terkoneksi dengan jaringan internet. [3]

Karakteristik utama yang dimiliki oleh *website* ialah halaman - halaman yang saling terhubung, dan juga dilengkapi oleh domain sebagai alamat (url) atau *World Wide Web* (www) serta hosting sebagai media yang menyimpan banyak data. *Website* dapat diakses melalui jaringan internet dengan platform yang disebut browser, seperti chrome, mozilla firefox, internet explorer (IE), opera dan sebagainya. [4]

Jenis Kategori *Website* : [3]

- 1) Web Statis adalah *website* yang memiliki halaman yang tidak berubah. Perubahan pada suatu halaman website dapat dilakukan secara manual dengan mengubah *source code* yang menjadi struktur dari *website* tersebut.
- 2) Web Dinamis adalah *website* yang terstruktur dan diperuntukan untuk diupdate serta dipantau sesering mungkin. Pada web dinamis umumnya disediakan halaman *backend* yang berguna melakukan perubahan konten dari *website* tersebut tanpa harus melakukan pengkodean ulang. Contohnya : web portal, web berita, dll.
- 3) Web Interaktif merupakan *website* yang dapat berinteraksi dengan penggunanya. Biasanya web ini berupa forum diskusi ataupun blog. Dimana terdapat moderator sebagai pengatur alur diskusi.

### 2.1.2 HTML

HTML (*Hyper Text Markup Language*) merupakan sebuah bahasa formatting yang dapat digunakan untuk membangun sebuah halaman *website*. Pada dunia pemrograman berbasis web (*Web Programming*), HTML merupakan sebuah pondasi dasar pada halaman *website*. File HTML di simpan dengan format .html (dot html). dan dapat di buka atau diakses dengan menggunakan web browser seperti Google Chrome, Mozilla Firefox, Opera, Safari dan lain-lain. Seperti yang sudah di jelaskan di atas, HTML merupakan dasar dari sebuah *website*. Untuk membuat *website* tidak cukup hanya menggunakan HTML, kita memerlukan bantuan CSS, JavaScript atau PHP untuk membangun suatu *website* dengan halaman yang dinamis, jika halaman website hanya dibangun dengan HTML maka halaman *website* tersebut di namakan halaman statis karena tidak memiliki aksi dan fungsi - fungsi yang dapat mengelola website, dengan kekurangan tersebut tentu developer akan sangat di sibukkan karena harus mengubah lagi file HTML setiap ingin mengupdate *website* tersebut. [5]

### 2.1.3 Mysql

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data (*Database Management Sistem*) atau DBMS yang *multi-thread, multi-user*, yang terdapat sekitar 6 juta instalasi didunia. MySQL merupakan sebuah *software* gratis yang berada di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual MySQL dibawah lisensi komersial untuk kasus - kasus yang dimana penggunaannya tidak sesuai dengan penggunaan GPL.

Tidak seperti PHP dan Apache yang merupakan sebuah *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulis masing - masing, MySQL dimiliki dan disponsori oleh perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta MySQL hampir atas semua kode sumbernya.

Kedua orang yang berasal dari Swedia dan satu orang berasal dari Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. [6]

#### **2.1.4 Metode Pengujian *Black Box***

Pengujian *black box*, juga disebut pengujian perilaku, yang berfokus pada persyaratan fungsional pada suatu perangkat lunak. *Black box testing* memungkinkan pengembang perangkat lunak untuk membuat set kondisi input yang sepenuhnya memenuhi semua persyaratan fungsional untuk sebuah program.

Pengujian *black box* atau pengujian fungsional adalah pengujian yang mengabaikan mekanisme internal sistem dan hanya berfokus pada output yang dihasilkan sebagai *respons* terhadap masukan dipilih dan kondisi kerjanya. Dari sini dapat disimpulkan bahwa *black box testing* adalah pengujian yang berorientasi pada fungsionalitas, yaitu perilaku perangkat lunak dihasilkan dari masukkan pengguna, sehingga keluaran yang tercapai tanpa melihat proses internal atau kode program yang dilakukan oleh perangkat lunak. [7]

*Black Box Testing* mencoba menemukan *bug* dalam kategori berikut:

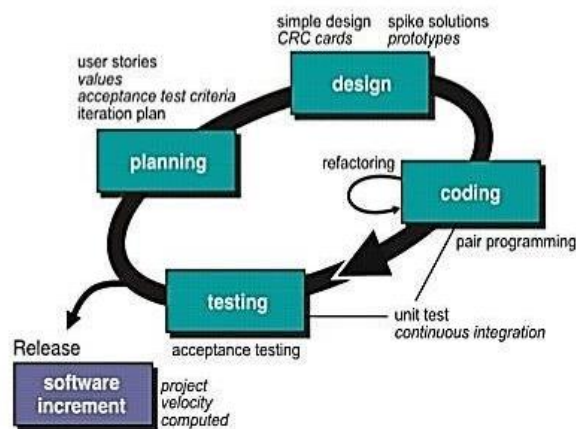
- 1) Fungsi salah atau hilang
- 2) Kesalahan UI
- 3) Kesalahan dalam struktur data (penggunaan basis data)
- 4) Kesalahan selama inisialisasi dan penghentian program
- 5) Kesalahan kinerja

#### **2.1.5 Metode Pengembangan Perangkat Lunak (*Extreme Programming*)**

*Extreme Programming* merupakan bagian dari *agile software development* yang paling umum digunakan. *Extreme Programming* (XP) adalah pengembangan perangkat lunak yang mengupayakan pendekatan berorientasi objek dan metode ini menargetkan tim yang dibuat dalam skala yang kecil hingga menengah dan metode ini juga cocok ketika tim dihadapkan dengan *requirement* yang tidak jelas, ataupun jika terdapat

perubahan - perubahan *requirement* yang sangat cepat. Metode XP menyederhakan berbagai fase pengembangan sistem informasi untuk membuatnya lebih efisien, mudah beradaptasi dan fleksibel. Nilai dasar *extreme programming* yaitu : *communication, Courage, Simplicity, Feedback dan Quality Work*. [8]

Berikut gambar model *Extreme Programming* :



**Gambar 2.1** Extreme Programming  
(Sumber : Mei Prabowo 2020)

XP menggunakan konsep pendekatan yang berorientasi pada objek dan juga memiliki empat kerangka kegiatan, yaitu :

### 1) *Planning*

Kegiatan Perencanaan dimulai dengan mengumpulkan *requirement* yang memungkinkan anggota teknis tim XP agar memahami konteks bisnis *software* dan mendapatkan gambaran lengkap tentang hasil yang diperlukan dan fungsionalitas perangkat lunak yang dibangun,

### 2) *Design*

Desain pada XP mengikuti prinsip KIS (“*Keep It Simple*”). Desain sederhana lebih disukai daripada desain yang rumit. Desain untuk fungsi tambahan tidak disarankan (karena *developer* akan diminta nanti). XP menggunakan CRC (*Class Responsibility Card*) sebagai

mekanisme efisien untuk pemikiran orientasi objek dalam perangkat lunak.

### 3) *Coding*

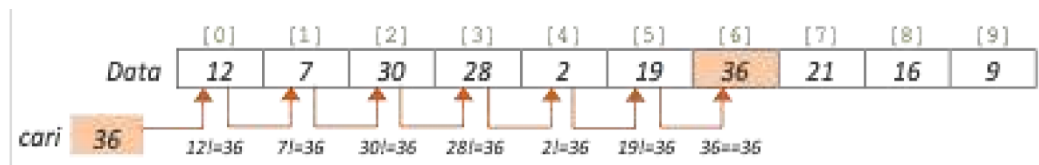
Setelah cerita selesai dikembangkan serta desain kerja awal diselesaikan, tim XP tidak langsung beralih ke tahap *coding*, melainkan mengembangkan serangkaian pengujian unit untuk dijalankan terhadap setiap cerita yang akan dibuat. Setelah tes unit dibuat, pengembang lebih fokus pada apa yang perlu dilakukan untuk lulus tes tersebut. Tidak ada sesuatu yang berlebihan untuk ditambahkan (prinsip KIS). Setelah kode siap, dapat langsung diuji, sehingga dapat memberikan umpan balik langsung kepada pengembang.

### 4) *Testing*

Setelah tahap *coding* selesai, kemudian dilakukan tahap pengujian sistem, dimana kita mengetahui eror apa saja yang terjadi selama aplikasi berjalan dan apakah sistem yang dibuat telah sesuai dengan kebutuhan pengguna.

## 2.1.6 Algoritma Pencarian Linier (*Linier Search*)

*Sequential Search* (pencarian berurutan) juga dikenal sebagai pencarian linier. Pencarian linier adalah salah satu algoritma pencarian paling sederhana dan paling dasar yang bekerja menggunakan pendekatan *bruteforce* dan mencoba berbagai cara untuk menemukan item yang anda inginkan. Misalnya, temukan nilai 36 dari data array {12, 7, 30, 28, 2, 19, 36, 21, 16, 9}. Pencarian linier mulanya dimulai dengan membandingkan/mencocokkan nilai 36 dengan elemen/index pertama [0] yaitu 12, karena 12 tidak sesuai dengan kriteria pencarian ( $12 \neq 36$ ) kemudian pencarian dilanjutkan dengan membandingkan nilai 36 dengan elemen berikutnya, sampai elemen yang cocok berhasil ditemukan. yaitu index [6] dimana  $36 == 36$ . Ilustrasi cara kerja pencarian linier dapat dilihat pada Gambar 2.2. [9]



**Gambar 2.2** Contoh Pencarian Linier  
(Sumber : Meidyan, Guntoro, Rezania 2022)

Implementasi algoritma pencarian linier dapat dilihat pada Gambar 2.2. *Sequential Search* dijelaskan pada kelas *Sequential Search* (baris 6) sedangkan pencarian data dilakukan didalam *method search* (baris 7). *Method search* akan menerima kunci pencarian dari pengguna (baris 12) lalu kemudian menggunakan *loop* (baris 13) untuk kemudian membandingkan kunci pencarian dengan semua elemen dalam array (baris 15). Program akan berhenti ketika elemen telah ditemukan (baris 16) atau ketika semua elemen dibandingkan (baris 20). Hasil program dapat dilihat pada Gambar 2.2.

Adapun Proses algoritma pencarian linier adalah sebagai berikut: [10]

- 1) Pertama data melakukan perbandingan berturut - turut satu per satu dengan data lain yang di cari dalam dataset sampai data ditemukan atau tidak ditemukan.
- 2) Pada dasarnya, pencarian ini hanya mengulang data dari 1 sampai dengan jumlah data (n).
- 3) Pada setiap pengulangan, dibandingkan data ke-1 dengan data yang akan dicari.
- 4) Jika data sesuai dengan yang dicari, berarti data berhasil di temukan. Disisi lain, jika sampai akhir pengulangan tidak ada data yang sama berarti data tersebut tidak ditemukan.

Urutan algoritma pencarian linier :

- 1)  $i \leftarrow 0$
- 2)  $Ketemu \leftarrow false$
- 3) Selama ( tidak ketemu ) dan (  $i < N$  ) kerjakan baris 4
- 4) Jika (  $Data[i] = key$  ) maka  $ketemu \leftarrow true$  Jika tidak  $i \leftarrow i + 1$
- 5) Jika (  $Ketemu$  ) maka  $i$  adalah indeks dari data yang dicari

### 2.1.7 UML (*Unified Modeling Language*)

UML adalah bahasa visual untuk permodelan dan komunikasi pada sebuah sistem dengan menggunakan diagram dan teks – teks pendukung. UML sendiri muncul karena adanya kebutuhan permodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak.

UML hanya digunakan untuk melakukan permodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. [11]

Contoh diagram Unified Modelling Language (UML) adalah sebagai berikut :

#### 1) *Use Case* Diagram

*Use case* diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* dapat digunakan untuk mengetahui fungsi - fungsi apa saja yang terdapat di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Syarat penamaan *use case* adalah nama yang didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*. [11]




Berikut adalah simbol – simbol dari diagram *use case* :

**Tabel 2.1** Simbol Diagram Use Case  
(Sumber : Rosa & Shalahuddin 2013)

Simbol	Nama	Deskripsi
	Use case	fungsi yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dapat dinyatakan dengan kata kerja di awal frase nama use case
	Aktor	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi meskipun simbol dari aktor merupakan gambar orang, tapi aktor belum tentu merupakan orang; biasanya dapat dinyatakan menggunakan kata benda di awal frase nama aktor
	Asosiasi	komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor



Tabel 2.1 (lanjutan)

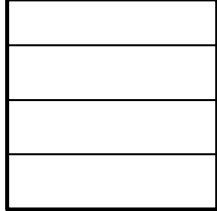
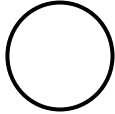



	Ekstensi	relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
	<<include>>  <uses>	<i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

## 2) Diagram Kelas (*Class Diagram*)



*Class diagram* menggambarkan struktur sebuah sistem dari segi pendefinisian kelas-kelas yang ingin dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak, sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancang dan hasil jadinya tidak sesuai. [11]

Berikut adalah simbol – simbol dari *class diagram* :

**Tabel 2.2** Simbol *Class Diagram*  
(Sumber : Rosa & Shalahuddin 2013)

Simbol	Nama	Deskripsi
	Kelas	Kelas pada struktur sistem
	Antarmuka	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
	Asosiasi	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Asosiasi berarah	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)

**Tabel 2.2** (lanjutan)


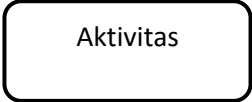
	Kebergantungan	Relasi antarkelas dengan makna kebergantungan antar kelas
	Agregasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)

### 3) Diagram Aktivitas (*Activity Diagram*)

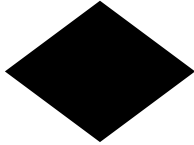


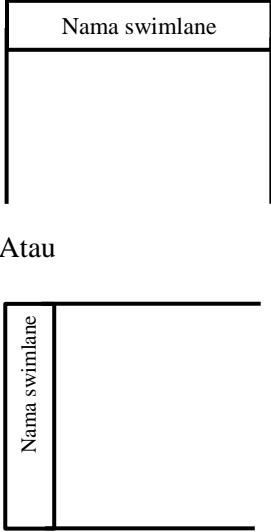
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. [11]

Berikut adalah simbol – simbol dari *Activity Diagram* :

**Tabel 2.3** Simbol *Activity Diagram*  
(Sumber : Rosa & Shalahuddin 2013)

Simbol	Nama	Deskripsi
	Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja

Tabel 2.3 (lanjutan)

	Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
 <p>Atau</p>	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

## 2.2 Penelitian Terkait

Berikut adalah daftar penelitian terkait yang menjadi referensi pada penelitian ini :

Tabel 2.4 Penelitian Terkait

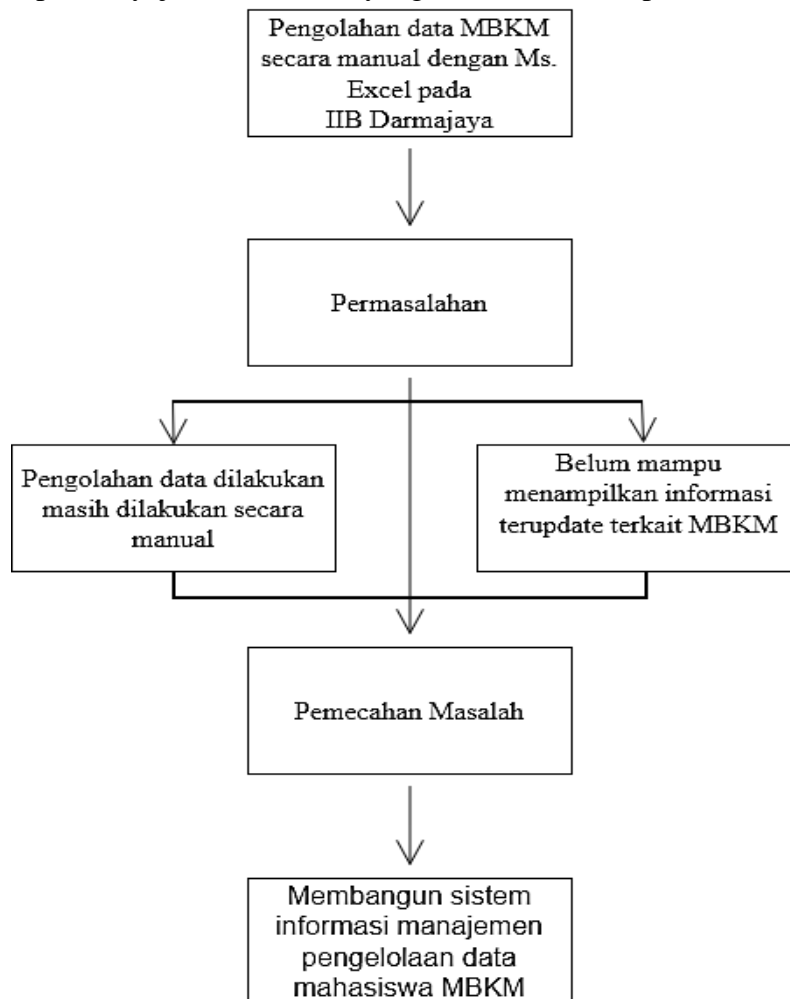
No	Nama Peneliti	Judul	Tahun	Uraian
1	Nani, Saktika, Agus, Kingking, Maman	Partisipasi civitas Akademik dalam Implementasi Merdeka Belajar Kampus Merdeka	2022	Tujuan dari penelitian ini adalah untuk menggambarkan kesiapan dan

Tabel 2.4 (lanjutan)

		(MBKM		kendala pelaksanaan program MBKM
2	Leon A. Abdillah	MBKM Berbasis Teknologi Informasi Sebagai Model Pendidikan Terkini	2021	Dengan sistem MBKM aktivitas mahasiswa diluar perguruan tinggi dapat dikonversi menjadi kredit belajar. Hal ini dilakukan untuk mengakomodir pengalaman dan pengetahuan belajar yang sebenarnya ada selain dikampus
3	Risma, Kustanto, Hasbi	Pencarian Data Barang Produk Atribut Sekolah Menggunakan Algoritma Binary Search	2021	Penerapan binary search bertujuan untuk mengatasi kendala pencarian data barang
4	T K Wulandari, E D Oktaviani, A Lestari	Penerapan Metode Binary Search dan Hamming Distance pada E-Library SMAN 2 Katingan Hilir	2022	E-Library mampu membuat pencatatan daftar buku menjadi lebih rapih, waktu untuk mencari buku lebih cepat, proses peminjaman dan pengembalian buku menjadi lebih terkomputerisasi
5	Dewi, Mike, Tri	Penggunaan Metode Binary Search pada Translator Bahasa Indonesia – Bahasa Jawa	-	Software ini dibuat dengan tampilan sederhana yang mampu memudahkan pengguna untuk mengaksesnya dan diharapkan mampu menambah minat baca khususnya para pelajar SD dan SMP

### 2.3 Kerangka Pemikiran

Alur pemikiran peneliti diterapkan dalam sebuah bagan mengenai sistem informasi manajemen pengelolaan data mahasiswa yang mengikuti program MBKM pada IIB Darmajaya. Pengolahan data yang masih dilakukan secara manual menggunakan microsoft excel menimbulkan suatu masalah seperti sulitnya untuk mengolah dan mencari data mahasiswa yang mengikuti program tersebut. Serta menyulitkan pihak kampus untuk menghitung jumlah mahasiswa yang mengikuti program tersebut. Untuk pemecah masalah yang ada dibutuhkan sebuah sistem informasi MBKM pada IIB Darmajaya yang diharapkan mampu menghasilkan sistem informasi secara menyeluruh yang mampu menyajikan informasi yang informatif dan up to date.



**Gambar2.3** Kerangka Pemikiran

## 2.4 Hipotesis

Hipotesis adalah dugaan atau pernyataan sementara yang digunakan untuk menyelesaikan suatu permasalahan. Dalam penulisan ini hipotesis dapat ditulis sebagai berikut :

- 1) Peneliti mampu merancang sistem informasi mahasiswa yang mengikuti program MBKM pada IIB Darmajaya.
- 2) Menerapkan algoritma pencarian linier kedalam sistem informasi.
- 3) Implementasi program sistem informasi mahasiswa yang mengikuti program MBKM pada IIB Darmajaya.

Hipotesis yang diajukan merupakan kemampuan peneliti dalam merencanakan, melaksanakan, dan mengimplementasikan program berdasarkan hasil yang akan didapat.