

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Aplikasi merupakan Teknologi yang berkembang pesat pada saat ini, kemajuan teknologi dalam membantu pengolahan data atau informasi yang tersedia dapat berlangsung secara cepat dan efisien serta akurat. Kemajuan aplikasi dapat dilihat dari banyaknya aplikasi yang dapat memudahkan penggunaannya. Salah satu aplikasi yang dapat memudahkan mahasiswa dalam belajar pemrograman adalah aplikasi pengembangan pembelajaran bahasa pemrograman.

Seperti halnya yang kita tahu, pembelajaran pemrograman merupakan salah satu pembelajaran penting di Teknik Informatika. Universitas Islam Sumatera Utara adalah salah satu Universitas yang memiliki jurusan Teknik Informatika. Mahasiswa UISU diharuskan untuk mengetahui bahasa pemrograman dari yang *low level* hingga *high level*. Mempelajari bahasa pemrograman dihimbau agar dapat mempermudah mahasiswa yang akan memilih skripsi pada peminatan pemrograman.

Bahasa pemrograman adalah satu set sintaks dan semantik yang digunakan untuk mendefinisikan program komputer. Bahasa pemrograman memungkinkan mahasiswa dapat menentukan data mana yang akan diproses oleh komputer, bagaimana data ini akan disimpan atau diteruskan, dan langkah-langkah apa yang akan diambil dalam berbagai situasi.

Banyaknya permasalahan mahasiswa dalam memahami bahasa pemrograman menjadi salah satu alasan saya membuat aplikasi ini. Kurangnya pemahaman mahasiswa dalam mempelajari bahasa pemrograman dapat mempersulit mahasiswa untuk membuat suatu aplikasi. Banyak mahasiswa yang hanya berpatok kepada pembelajaran tatap muka, sehingga membuat pemahaman mereka kurang dalam pemrograman.

Memperhitungkan persentase kemampuan mahasiswa dapat dilakukan dengan membuat statistik kemampuan. Pada aplikasi ini saya akan membuat statistik kemampuan untuk mengetahui kemampuan-kemampuan mahasiswa tentang bahasa pemrograman yang ada.

Aplikasi ini saya buat untuk memperkenalkan lebih detail tentang beberapa bahasa pemrograman untuk membantu mahasiswa dalam mempelajari bahasa pemrograman itu sendiri. Aplikasi ini juga untuk mempermudah dosen di Universitas Islam Sumatera Utara untuk mengetahui sejauh mana mahasiswa mengetahui tentang bahasa pemrograman yang ada. Maka, dengan adanya masalah-masalah yang ada, saya membuat aplikasi ini dengan judul **“PENERAPAN ALGORITMA *NAÏVE BAYES* DAN *SELECTION SORT* PADA APLIKASI PEMBELAJARAN PEMROGRAMAN JAVA DAN PHP”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, maka dapat dirumuskan beberapa masalah sebagai berikut :

1. Bagaimana pengembangan pembelajaran bahasa pemrograman untuk mahasiswa yang dapat dipahami dengan baik?
2. Bagaimana cara membangun aplikasi untuk pengembangan pembelajaran bahasa pemrograman?
3. Bagaimana penggunaan algoritma *Naive Bayes* dan *Selection Sort* dapat berjalan pada aplikasi pembelajaran pemrograman Java dan PHP?
4. Bagaimana statistik kemampuan dapat berjalan?

1.3 Batasan Masalah

Adapun beberapa batasan masalah yang ada dalam proposal skripsi ini adalah sebagai berikut :

1. Aplikasi ini menggunakan bahasa PHP, HTML dan database MySQL.
2. Aplikasi ini menggunakan data dari mahasiswa program studi Teknik Informatika berjumlah 10 orang dari tahun ajaran 2020 dan 2021.
3. Aplikasi ini menggunakan sesi kuis sebagai bahan untuk mendapatkan data statistik kemampuan.
4. Aplikasi ini merupakan aplikasi *website*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Untuk mengembangkan pembelajaran bahasa pemrograman mahasiswa di Teknik Informatika dengan belajar bahasa pemrograman satu per satu dengan belajar setiap part yang ada didalam bahasa pemrograman tersebut.

2. Untuk membangun aplikasi pengembangan pembelajaran bahasa pemrograman mahasiswa di Teknik Informatika dengan menggunakan PHP dan MySql.
3. Untuk mengetahui persentase pengetahuan mahasiswa Teknik Informatika UISU dengan menggunakan statistik kemampuan yang akan dihitung dari data yang masuk dengan menyelesaikan quis yang ada di tiap bahasa pemrograman.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah untuk membuat aplikasi berbasis web yang dapat digunakan sebagai media pembelajaran mahasiswa dalam mempelajari bahasa pemrograman yang ada. Mahasiswa akan diminta untuk mengisi kuis yang ada untuk mengetahui persentase kemampuan mahasiswa di prodi Teknik Informatika Universitas Islam Sumatera Utara.

1.6 Metode Penelitian

Metode penelitian yang digunakan pada aplikasi ini adalah observasi, wawancara, dan studi kasus.

1.7 Sistematika Penulisan

Untuk memperoleh gambaran yang utuh untuk penulisan ini, maka diperlukan sistematika penulisannya yang akan dibagi menjadi lima bab dengan rincian sebagai berikut :

BAB I : PENDAHULUAN

Merupakan bab pendahuluan yang berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Dalam bab ini akan menguraikan mengenai teori-teori yang ada.

BAB III : METODE PENELITIAN

Pada bab ini diuraikan tentang gambaran objek penelitian dan analisis dibutuhkan dalam pembuatan sistem . laporan secara detail rancangan terhadap penelitian yang dilakukan, baik perancangan secara umum dari sistem yang dibangun maupun perancangan yang lebih spesifik.

BAB IV : HASIL DAN PEMBAHASAN

Dalam Bab ini membahas tentang hasil yang telah dicapai maupun masalah-masalah yang ditemui selama penelitian, uji coba, termasuk kelemahan dan kelebihan sistem yang dibuat.

BAB V : PENUTUP

Bab ini menjelaskan kesimpulan dan saran dari laporan skripsi yang penulis buat.

BAB II

TINJAUAN PUSTAKA

2.1 Algoritma

2.1.1 Pengertian Algoritma

Algoritma adalah sistim kerja komputer memiliki *brainware*, *hardware*, dan *software*. Tanpa salah satu dari ketiga sistim tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada *software* komputer. Software terbangun atas susunan program) dan *syntax* (cara penulisan/pembuatan program). Untuk menyusun program atau *syntax*, diperlukannya langkah- langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu *software*. Maka, algoritma berperan penting dalam penyusunan program atau *syntax* tersebut.

Pengertian algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu *software*. Dalam dunia sehari-hari, mungkin tanpa kita sadari algoritma telah masuk dalam kehidupan kita.

2.1.2 Strukur Dasar Algoritma

Adapun struktur dasar pada algoritma adalah sebagai berikut:

a. Sekuensial (runtunan)

Pada struktur sekuensial ini langkah-langkah yang dilakukan dalam algoritma diproses secara berurutan. Dimulai dari langkah pertama, kedua, dan seterusnya. Pada dasarnya suatu program memang menjalankan suatu proses dari yang dasar seperti struktur ini.

b. Struktur seleksi

Struktur seleksi menyatakan pemilihan langkah yang didasarkan oleh suatu kondisi atau pengambilan suatu keputusan. Struktur ini ditandai selalu dengan bentuk *flowchart decision* (*flowchart* yang berbentuk belah ketupat).

Banyak contoh yang dapat kita terapkan pada struktur jenis ini jika itu menyangkut keputusan, diantaranya: diskon yang berbeda berdasarkan jumlah barang yang ingin dibeli.

c. Struktur perulangan

Struktur ini memberikan suatu perintah atau tindakan yang dilakukan beberapa kali. Misalnya jika teman mau menuliskan kata “belajar c” sebanyak sepuluh kali. Akan lebih efisien jika teman menggunakan struktur ini dari pada sekedar menuliskannya berturut-turut sebanyak sepuluh kali

2.2 Algoritma *Naive Bayes*

Algoritma *Naive Bayes* (NB) dapat digunakan untuk masalah klasifikasi biner dan multikelas. NB membuat dan menilai model dengan sangat cepat dan skala secara linear dalam jumlah prediksi dan baris. NB juga merupakan

klasifikasi yang mempresetasikan setiap kelas objek berdasarkan kesimpulan atau rekapitulasi probabilistik dan menemukan kemungkinan besar kelas yang sesuai dengan tiap objek yang akan ditentukan kelasnya dari objek-objek uji yang ada berdasarkan atribut-atribut atau variabel yang telah ditentukan nilainya.

Klasifikasi NB juga tergolong ke dalam pembelajaran yang terawasi (*supervised learning*) dimana untuk menentukan kelas objek uji berdasarkan pada data latihan dan setiap objek didata latihan telah diketahui kelasnya masing-masing. NB mengasumsikan bahwa fitur-fiturnya atau variabel atau atributnya adalah independen sesuai kelas yang ditentukan, artinya bahwa semua fitur hanya mempunyai kelas sebagai node asal atau node orang tua (Kononenko, 1990; Langley, 1992; Domingos, 1997; Mitchell, 1997). Penggunaan NB untuk klasifikasi dianggap penting karena beberapa hal, seperti (Wu. X., dll., 2007);

- a. Sangat mudah untuk dibangun karena tidak memerlukan skema untuk estimasi parameter iterative yang rumit dan metode ini dapat langsung diimplementasikan ke dalam jumlah data dalam skala yang sangat besar.
- b. Mudah untuk ditafsirkan sehingga pengguna yang kurang terampil dalam teknik klasifikasi dapat dengan mudah memahami hasil akhir yang diperoleh.

Namun demikian, walaupun klasifikasi NB mudah dibangun dan bekerja dengan baik, tetapi dalam beberapa domain NB bekerja kurang baik, hal ini disebabkan oleh performa NB yang membiarkan adanya korelasi atau tidak relevannya atribut-atributnya.

Jika dua atau lebih atribut yang memiliki korelasi sangat kuat, mereka menerima terlalu banyak bpbpt dalam keputusan akhir terhadap kelas mana yang akan ditempati oleh suatu objek.

Klasifikasi NB didasarkan pada teorema *bayes* dengan asumsi adanya sifat independen antara atribut atau variabel yang ada, atau dengan kata lain setiap atribut, fitur atau variabel yang ada tidak saling bergantung antara satu dengan yang lain. Selain itu, dalam NB ini juga diasumsikan bahwa semua fitur atau variabel sama pentingnya, tetapi asumsi-asumsi tersebut secara umum boleh jadi tidak benar, tetapi secara praktis algoritmanya memberikan hasil yang sangat baik. Sebelum membahas lebih jauh tentang metode NB maka ada beberapa hal yang perlu di-*review*, seperti tentang probabilitas bersyarat dan teorema *bayes*. (Arhami, 2020)

Misalkan X merupakan sampel data dengan label kelasnya yang belum diketahui, dan misalkan H adalah beberapa hipotesis sehingga sampel data X ada dalam kelas C yang spesifik. Probabilitas bersyarat atau probabilitas H didalam X adalah probabilitas interaksi H dan X dari probabilitas X , atau dengan notasi lain dapat ditulis $P(H|X)$ yang merupakan persentase banyaknya H didalam X , atau dapat ditulis:

$$P(H|X) = \frac{P(H \cap X)}{P(X)} \dots\dots\dots(1)$$

Oleh karena itu, untuk menentukan $P(H|X)$ yang merupakan persentase banyaknya H dalam X . $P(H|X)$ adalah peluang posterior yang merepresentasikan tingkat kepercayaan. Rumus teorema *bayes* dapat dilihat seperti ini:

$$P(H|X) = \frac{P(X|H)*P(H)}{P(X)} \dots\dots\dots(2)$$

Dengan:

H = Hipotesisnya

X = Kejadian yang merelasikannya ke hipotesis

P(H) = Peluang hipotesis

P(X) =Peluang kejadian

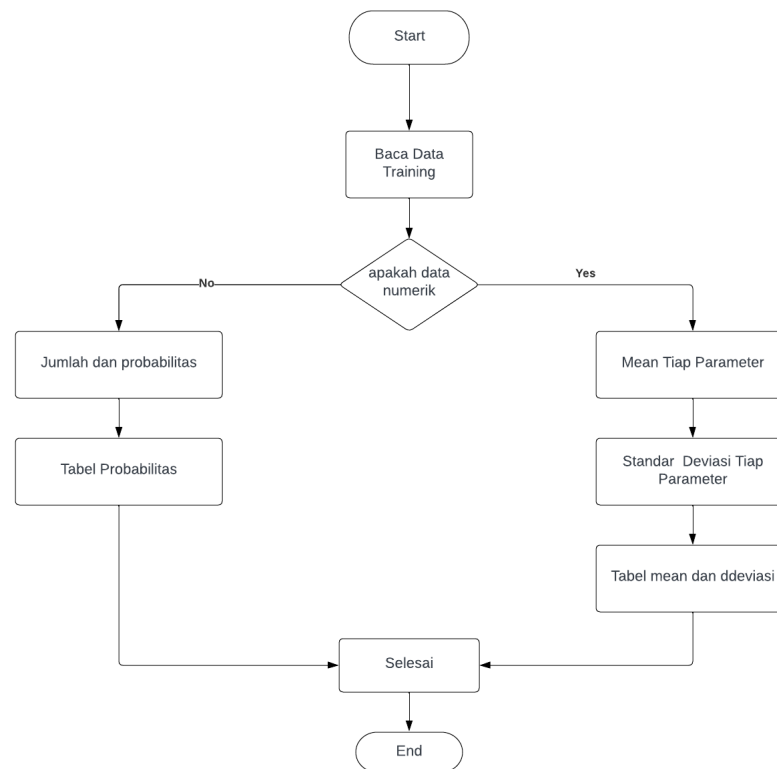
P(X|H) = Peluang banyaknya X didalam H

P(H|X) = Peluang banyaknya H didalam X

2.2.1 Flowchart Algoritma Naive Bayes

Algoritma *Naive Bayes* merupakan salah satu algoritma data *mining* yang sangat mempermudah dalam pembuatan suatu data numerik maupun non numerik.

Berikut *flowchart* contoh bagaimana algoritma *Naive Bayes* bekerja:



Gambar 2.1 *Flowchart* Cara Kerja *Naive Bayes*

Pada gambar 2.1 merupakan contoh bagaimana cara kerja algoritma *naive bayes*. Algoritma ini dimulai dengan membaca data *training* yang dimasukkan, kemudian algoritma ini akan melihat apakah data yang dimasukkan merupakan data numerik atau tidak. Jika data yang dimasukkan merupakan bilangan numerik, maka tahap awal yang dilakukan adalah menghitung nilai rata-rata dari setiap parameter, lalu menghitung standar deviasi tiap parameter dan menampilkan tabel rata-rata dan deviasi. Tetapi, jika data yang dimasukkan bukan bilangan numerik, maka algoritma ini akan menghitung jumlah probabilitas dan setelah itu menampilkan tabel probabilitas.

2.3 *Selection Sort*

2.3.1 Pengertian *Selection Sort*

Selection Sort merupakan salah satu algoritma pengurutan yang sederhana. Ide dasarnya adalah melakukan beberapa kali pass untuk melakukan penyeleksian elemen struktur data. Untuk *sorting ascending* (menaik) elemen yang paling kecil di antara elemen-elemen yang belumurut, disimpan indeksinya, kemudian dilakukan pertukaran nilai elemen dengan indeks yang disimpan tersebut dengan elemen yang paling depan yang belumurut. Sebaliknya, untuk *sorting descending* (menurun) elemen yang paling besar yang disimpan indeksinya kemudian ditukar. *Selection Sort* diakui karena kesederhanaan algoritmanya dan performanya lebih bagus dari pada algoritma lain yang lebih rumit dalam situasi tertentu. Algoritma ini bekerja sebagai berikut:

1. Mencari nilai minimum (jika *ascending*) atau maksimum (jika *descending*) dalam sebuah list.
2. Menukarkan nilai ini dengan elemen pertama list.
3. Mengulangi langkah di atas untuk sisa list dengan dimulai pada posisi kedua.

Secara efisien kita membagi list menjadi dua bagian yaitu bagian yang sudah diurutkan, yang didapat dengan membangun dari kiri ke kanan dan dilakukan pada saat awal, dan bagian list yang elemennya akan diurutkan. Berikut gambaran dari implementasi *Selection Sort*: 70, 60, 30, 50, 40,20.

Pass 1:

(70, 60, 30, 50, 40,20)

(20, 60, 30, 50, 40, 70) -> Geser (70,20)

Pass 2:

(20, 60, 30, 50, 40, 70)

(20, 30, 60, 50, 40, 70) -> Geser (60,30)

Pass 3:

(20, 30, 60, 50, 40, 70)

(20, 30, 40, 50, 60, 70) -> Geser (60,40)

Pass 4:

(20, 30, 40, 50, 60, 70)

(20, 30, 40, 50, 60, 70) -> Tidak ada proses pergeseran karena nilai pada pass 4 sudah tepat tidak ada yang lebih kecil dari pass tersebut.

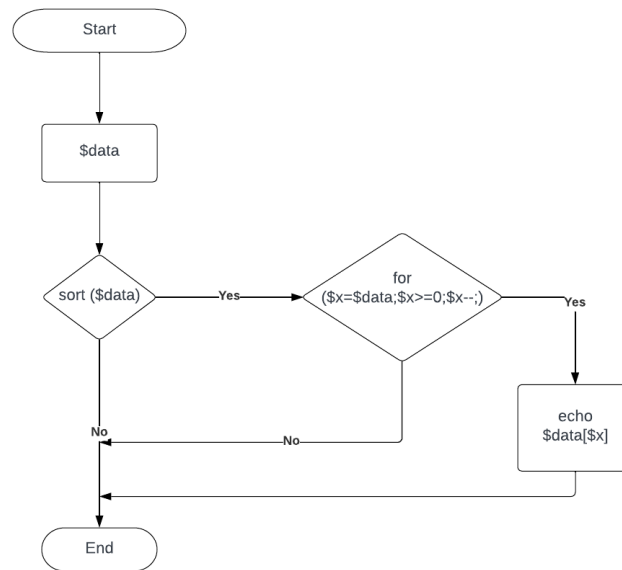
Pass 5:

(20, 30, 40, 50, 60, 70)

(20, 30, 40, 50, 60, 70) -> Tidak ada pergeseran pada pass 5 karena nilai sudah tepat

Jika semua pass sudah memiliki nilai yang tepat maka proses *sorting* akan berhenti. Maka dari proses *sorting* diatas akan menghasilkan nilai: 20, 30, 40, 50, 60.

2.3.2 Flowchart Selection Sort



Gambar 2.2 Contoh Flowchart Selection Sort

Pada gambar 2.2 merupakan contoh rangkaian cara kerja *selection sort*, dari data *flowchart* diatas akan menghasilkan sebuah *sorting* dengan keadaan *descending* (menaik) ke *ascending* (menurun). *Flowchart* tersebut dimulai dengan memperkenalkan variabel data yang lalu variabel data tersebut akan dimulai pengurutan dengan fungsi *sort*. Jika keadaan tersebut benar akan diurutkan, maka proses akan otomatis melanjutkan proses *sorting* dengan keadaan untuk variabel $x = \text{variabel data}$, dimana variabel x tidak sama dengan 0 dan perulangan dilakukan secara menurun. Maka hasil yang akan keluar nantinya adalah penurunan bilangan dari yang terbesar ke yang terecil.

2.4 Aplikasi

2.4.1 Pengertian Aplikasi

Aplikasi adalah program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus (Kadir, 2003).

Menurut Kadir (2008:3) program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Dari kedua pengertian di atas dapat disimpulkan bahwa aplikasi adalah sekumpulan perintah atau kode yang disusun secara sistematis untuk menjalankan suatu perintah yang diberikan oleh manusia melalui komponen atau *hardware* komputer yang digunakan oleh manusia dalam menjalankan program aplikasi, dengan demikian bisa membantu manusia untuk memberikan solusi dari apa yang diinginkan.

Pengertian aplikasi menurut para ahli adalah sebagai berikut :

- a. Menurut Supriyanto (2005:2) adalah program yang memiliki aktifitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu.
- b. Menurut Jogiyanto (1999:12), Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*intruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.
- c. Menurut Dhanta (2009:32), aplikasi (*application*) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel.
- d. Menurut Anisyah (2000:30), aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

2.5 Pengembangan Pembelajaran

2.5.1 Pengertian Pengembangan

Pengembangan adalah suatu usaha untuk meningkatkan kemampuan teknis, teoritis, konseptual, dan moral sesuai dengan kebutuhan melalui pendidikan dan latihan. Pengembangan adalah suatu proses mendesain pembelajaran secara logis, dan sistematis dalam rangka untuk menetapkan segala sesuatu yang akan dilaksanakan dalam proses kegiatan belajar dengan memperhatikan potensi dan kompetensi peserta didik.

2.5.2 Pengembangan Pembelajaran

Pengembangan pembelajaran lebih realistik, bukan sekedar idealisme pendidikan yang sulit diterapkan dalam kehidupan. Pengembangan pembelajaran adalah usaha meningkatkan kualitas proses pembelajaran, baik secara materi maupun metode dan substansinya. Secara materi, artinya dari aspek bahan ajar yang disesuaikan dengan perkembangan pengetahuan, sedangkan secara metodologis dan substansinya berkaitan dengan pengembangan strategi pembelajaran, baik secara teoritis maupun praktis.

2.6 Program

2.6.1 Pengertian Program

Program merupakan kumpulan instruksi yang digunakan untuk mengatur komputer agar dapat menjalankan tindakan tertentu. Jika tanpa program, komputer sesungguhnya tidak bisa berbuat apa-apa atau tidak akan berfungsi. Hanya mesin menjadi kosong. Selain itu, Program ialah salah satu bagian dari beberapa aspek penting dari komputer. Orang yang membuat program sering disebut sebagai pemrograman atau programmer.

Aktivitas membuat program disebut sebagai pemrograman. Jadi pemrograman merupakan suatu kumpulan urutan perintah ke komputer untuk mengerjakan sesuatu. Perintah-perintah ini membutuhkan suatu bahasa tersendiri. Sebagaimana bahasa manusia, bahasa pemrograman memiliki kaidah tertentu yang dapat dimengerti oleh komputer.

Pengertian Program Menurut Para Ahli

Berikut ini adalah pengertian yang di definisikan oleh para ahli. Mari kita simak bersama penjelasannya dibawah ini.

1. Sukrisno (2005), Program merupakan ekskresi, kata, atau pernyataan yang dirangkai dan disusun menjadi satu kesatuan prosedur, yang berupa urutan langkah-langkah, untuk dapat menyelesaikan masalah yang diberikan dengan memakai bahasa pemrograman sehingga bisa dieksekusi oleh komputer.
2. Sugiyono (2005:21), Program adalah rangkaian instruksi-instruksi pada bahasa komputer yang disusun dengan sistematis dan logis.
3. Sunarto, S.Kom (2008), Program merupakan sekumpulan interaksi yang diwujudkan pada bentuk bahasa, kode skema, maupun lainnya, jika digabungkan dengan media yang bisa dibaca dengan komputer akan mampu membuat komputer bekerja agar dapat melakukan fungsi khususnya, dan juga persiapan dalam merancang interaksi tertentu.
4. Wiryanto Dewobroto (2004), Program merupakan sebuah hasil penyusunan detail langkah-langkah solusi (algoritma) suatu masalah.
5. Binanto (2009:1), Menurut Binanto, pemrograman dapat diartikan dalam beberapa hal, sebagai berikut:
 - Mendeskripsikan instruksi-instruksi tersendiri yang biasanya disebut sebagai Source Code yang dibuat oleh programmer.
 - Mendeskripsikan suatu keseluruhan bagian dari software yang executable.
 - Program merupakan himpunan atau kumpulan instruksi tertulis yang dibuat oleh programmer atau suatu bagian executable dari suatu software.
 - Pemrograman berarti membuat program komputer.

- Pemrograman merupakan suatu kumpulan urutan perintah ke komputer untuk mengerjakan sesuatu.
 - Perintah-perintah ini membutuhkan suatu bahasa tersendiri yang dapat dimengerti oleh komputer.
6. Wikipedia, Program komputer merupakan sekumpulan instruksi yang ditulis untuk melaksanakan fungsi spesifik pada komputer.

2.6.2 Fungsi Program

Adanya program menjadi suatu hal yang harus ada pada sebuah komputer. Karena program bisa mengontrol *Hardware* (perangkat keras) untuk menjalankan fungsinya dengan baik. Tanpa program, komputer hanyalah sebuah mesin yang tidak dapat melakukan tugas apapun yang diinginkan oleh penggunanya.

Dari penjelasan diatas sudah bisa kita ambil kesimpulan bahwa fungsi utama program adalah mengendalikan komputer agar dapat melakukan tugas spesifik seperti yang diinginkan oleh pengguna. Program juga memiliki fungsi lain diantaranya :

1. Program berfungsi mengatur hubungan antar hardware agar dapat bekerja secara bersamaan. Dengan demikian, *hardware* bisa bekerja sesuai dengan fungsinya masing-masing tanpa ada hambatan.
2. Program berfungsi untuk mengidentifikasi program. Misalnya, File Explorer yang berperan menjadi tempat untuk mengakses berbagai jenis program lainnya yang tersimpan dalam komputer.

3. Program berfungsi untuk menghubungkan antara program dengan *hardware*. Contohnya, *Device Manager* yang berfungsi sebagai program dalam mengelola seluruh *hardware* yang dikenali oleh sistem operasi *Windows*.
4. Program berfungsi untuk menerjemahkan instruksi kedalam bahasa mesin. Contohnya seperti compiler dari bahasa pemrograman, seperti Visual Basic, Java, C++ dan lain-lainnya.

2.7 Bahasa Pemrograman

2.7.1 Pengertian Bahasa Pemrograman

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa yang akan diambil dalam berbagai situasi secara persis.

Dari sekian bahasa yang digunakan oleh *programmer*, terdapat beberapa bahasa yang memiliki *trending* terbaik menurut StackOverflow, mereka menyebutnya “*Most Popular Technology*”. Artikel ini akan membahas 5 dari beberapa bahasa pemrograman terpopuler tersebut yang bisa menjadi pilihan kamu untuk memulai belajar bahasa pemrograman.

2.8 Java

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama Sun Microsystems, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “Oak”, namun pada tahun 1995 diganti namanya menjadi “Java”.

2.8.1 Pengertian Java

Menurut definisi Sun Microsystem, di dalam buku M. Shalahuddin dan Rosa A.S. (2010 : 1) Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*stand alone*) ataupun pada lingkungan jaringan. Java berdiri di atas sebuah mesin penterjemah (interpreter) yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca kode bit (*bytecode*) dalam file *.class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada system operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada *platform* (*platform independent*). Itulah yang menyebabkan dalam dunia pemrograman Java dikenal adanya istilah “*write once, run everywhere*”, yang berarti kode program hanya ditulis sekali, namun dapat 9 dijalankan di bawah kumpulan pustaka (*platform*) manapun, tanpa harus melakukan perubahan kode program.

2.8.2 Arsitektur Java

Secara arsitektur, Java tidak berubah sedikitpun sejak awal mula bahasa tersebut dirilis. *Compiler* Java (yang disebut dengan `javac` atau *Java Compiler*) akan mentransformasikan kode-kode dalam bahasa Java ke dalam suatu kode bit. Dimana *bytecode* adalah sekumpulan perintah hasil kompilasi yang kemudian dapat dieksekusi melalui sebuah mesin komputer abstrak, yang disebut dengan JVM (*Java Virtual Machine*). JVM juga sering dinamakan sebagai interpreter, karena sifatnya yang selalu menerjemahkan kode-kode yang tersimpan dalam kode bit dengan cara baris demi baris. Untuk menjalankan program Java, maka file dengan ekstensi `.java` harus dikompilasi menjadi file kode bit. Dimana untuk menjalankan kode bit tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan pustaka Java yang digunakan.

2.9 PHP

2.9.1 Sejarah PHP

Pada awalnya PHP digunakan sebagai bahasa pemrograman untuk sebuah *server-side HTML-embedded* dengan nama *Personal Home Pages*. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessing/Form Interpreter*. Dengan perilisannya ini menjadi *open*

source, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0. Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2.9.2 Keunggulan PHP

Beberapa keunggulan yang ditawarkan PHP dibandingkan beberapa bahasa pemrograman lainnya adalah :

≡ Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.

≡ Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.

≡ Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.

≡ Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.

≡ PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.

2.10 SQL

Structured Query Language atau SQL adalah sebagai sebuah bahasa khusus yang digunakan ketika membuat maupun mengolah database dalam sebuah sistem. Profesi yang menangani bahasa ini biasanya adalah *Back-End Developer* dan *Full Stack Developer*.

2.10.1 Sejarah MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang kala itu bernama TcX DataKonsult AB, sejak sekitar 1994–1995, meski cikal bakal kodenya bisa disebut sudah ada sejak 1979. Tujuan mula-mula TcX membuat MySQL pada waktu itu juga memang untuk mengembangkan aplikasi Web untuk klien—TcX adalah perusahaan pengembang software dan konsultan database. Kala itu Michael Widenius, atau “Monty”, pengembang satu-satunya di TcX, memiliki aplikasi UNIREG dan rutin ISAM yang dibuat sendiri dan sedang mencari antarmuka SQL untuk ditempelkan di atasnya. Mula-mula TcX memakai mSQL, atau “mini SQL”. Barangkali mSQL adalah satu-satunya kode database open source yang tersedia dan cukup sederhana saat itu, meskipun sudah ada

Postgres. Namun ternyata, menurut Monty, mSQL tidaklah cukup cepat maupun fleksibel. Versi pertama mSQL bahkan tidak memiliki indeks. Setelah mencoba menghubungi David Hughes—pembuat mSQL—dan ternyata mengetahui bahwa David tengah sibuk mengembangkan versi dua, maka keputusan yang diambil Monty yaitu membuat sendiri mesin SQL yang antarmukanya mirip dengan mSQL tapi memiliki kemampuan yang lebih sesuai kebutuhan. Lahirlah MySQL.

2.11 Pemodelan *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah sebuah bahasa pemodelan grafis yang digunakan sebagai standar untuk memodelkan sistem dengan metodologi pemodelan berorientasi objek (Mulyani, 2016). Sedangkan menurut (Mujilan, 2017) UML merupakan diagram sistem menggunakan orientasi objek (*object oriented*) dalam analisis dan perancangan sistem yang saat ini menjadi standar dalam berbagai tipe solusi perangkat lunak dalam pengembangan sistem.

Beberapa fungsi dan kegunaan dari UML yaitu (Mulyani, 2016:) :

1. *Visualizing*, yaitu sebagai alat komunikasi konseptual model antara tim pengembang sistem (sistem analis dengan programmer)
2. *Specifying*, yaitu sebagai tools yang digunakan untuk memodelkan sistem secara tepat dan jelas.
3. *Constructing*, yaitu UML sebagai bahasa grafis mampu melakukan mapping dan konseptual model kedalam bahasa pemrograman.
4. *Documenting*, yaitu UML digunakan sebagai tools untuk melakukan dokumentasi teknis sebuah sistem.

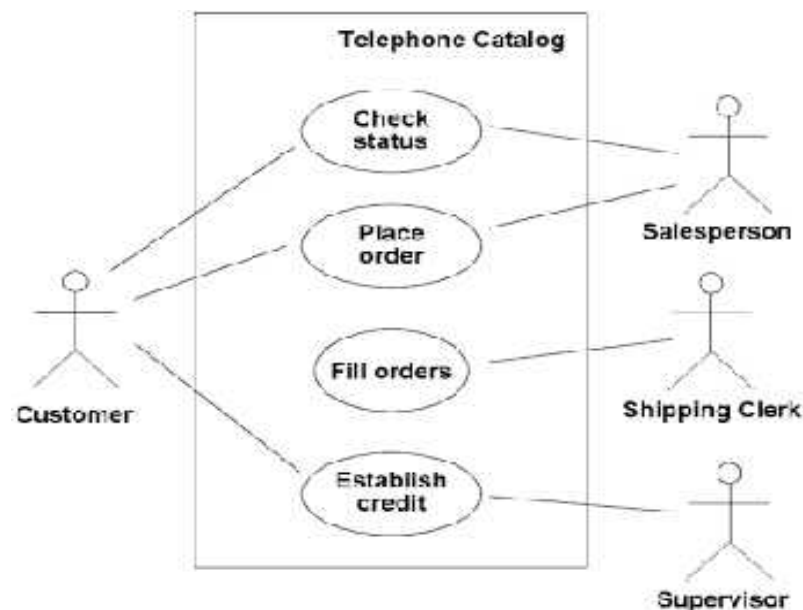
Terdapat beberapa diagram *Unified Modeling Language* (UML) yang sering digunakan dalam pengembangan sistem menurut (Mulyani, 2016) yaitu :

1. *Use Case Model*

Use case model merupakan kumpulan diagram dan *text* yang saling bekerja sama untuk mendokumentasikan bagaimana *user* (aktor) berinteraksi dengan sistem. *Use case model* terdiri dari beberapa diagram yaitu :

a. *Use Case Diagram*

Use case diagram yaitu diagram yang menggambarkan dan merepresentasikan aktor, *user cases* dan *dependencies* suatu proyek dimana tujuan dari diagram ini adalah untuk menjelaskan konsep hubungan antara sistem dengan dunia luar.



Gambar 2.3 Contoh *Use Case Diagram*

b. *Use Case Narrative*

Use case narrative yaitu deskripsi yang menjelaskan *use case diagram*. Pada *use case diagram* sistem hanya digambarkan secara

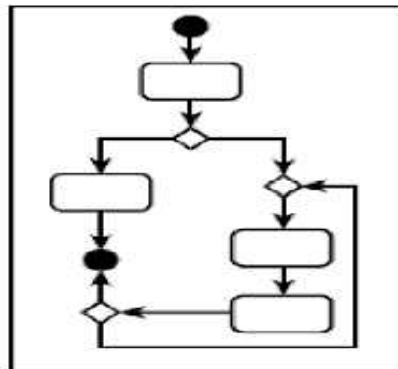
sederhana menggunakan simbol *use case* yang berhubungan (*relationship*) dengan aktor, sehingga terkadang diperlukan deskripsi yang menjelaskan dari proses tersebut.



Gambar 2.4 Contoh *Use Case Narrative*

c. *Use Case Skenario*

Use case scenario yaitu pemecahan kemungkinan logika pada *use case diagram*.

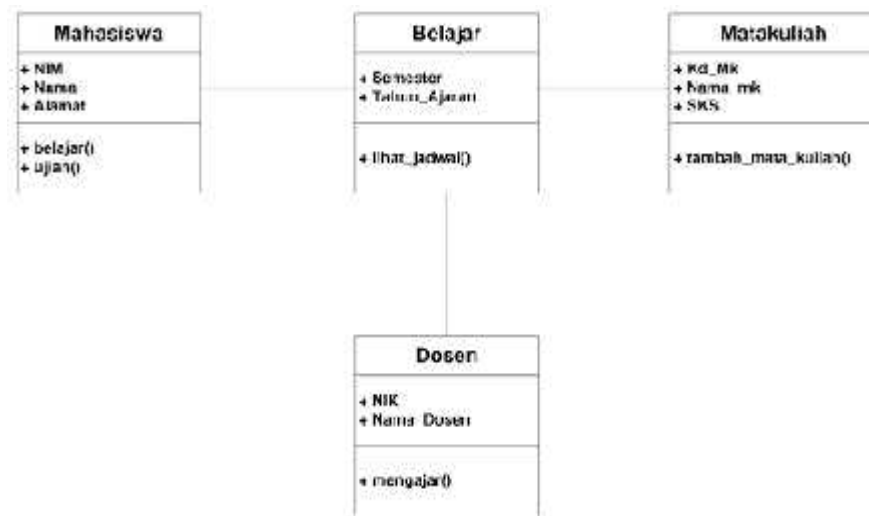


Gambar 2.5 Contoh *Use Case Skenario*

2. *Class Diagram*

Class diagram adalah diagram yang digunakan untuk merepresentasikan kelas, komponen-komponen kelas dan hubungan antar masing-masing kelas. Selain itu *class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga

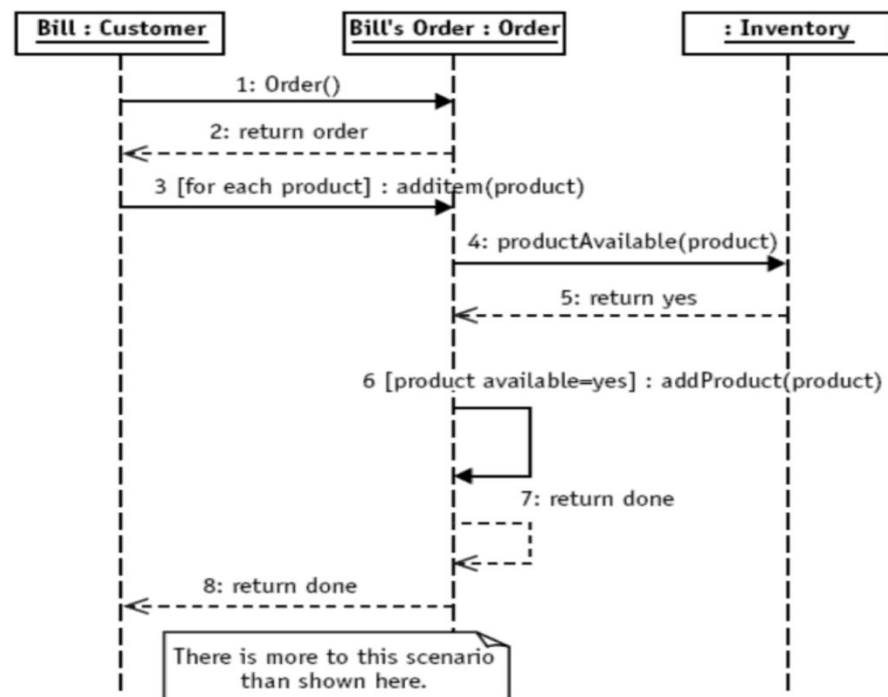
menunjukkan *property* dan operasi sebuah kelas serta batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.



Gambar 2.6 Contoh *Class Diagram*

3. *Activity Diagram*

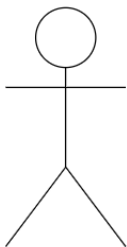
Activity diagram adalah diagram *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur aktivitas dari satu proses. *Activity diagram* memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya, dengan kata lain diagram hanya menyebutkan aturan-aturan rangkaian dasar yang harus diikuti.

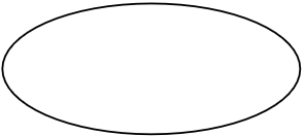


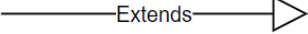
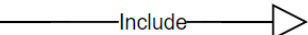
Gambar 2.7 Contoh *Sequence Diagram*

2.12 Daftar Simbol Diagram

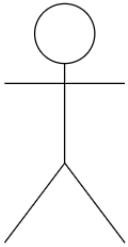
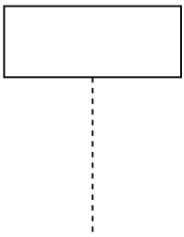



a. Daftar Simbol *Use Case Diagram*

Tabel 2.1 Tabel Daftar Simbol *Use Case Diagram*(Maharani, 2018)

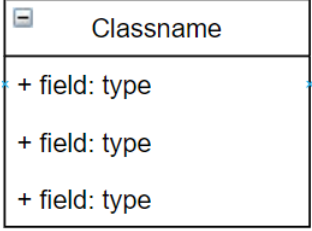



Simbol	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran ketika berinteraksi dengan sistem usulan.

	Use Case	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem, dan mewakili sebagian besar sistem secara fungsional.
	Sistem	Menggambarkan ruang lingkup sistem.
	Asosiasi	Menghubungkan aktor dengan use case yang berinteraksi.
	Ekstend	Relasi yang menggambarkan bahwa sebuah use case (sub use case) bisa berdiri sendiri atau bisa berjalan tanpa menjalankan main use case terlebih dahulu.
	Include	Relasi yang menggambarkan bahwa sebuah use case (sub use case) harus menjalankan use case lain terlebih dahulu sebelum menjalankan fungsinya.

b. Daftar Simbol *Sequence Diagram*Tabel 2.2 Tabel Daftar Simbol *Sequence Diagram*(Maharani, 2018)

Simbol	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran ketika berinteraksi dengan sistem usulan
	Object Lifeline	Menyatakan hidup uatu object dalam basis waktu
	Activation	Menyatakan object dalam keadaan aktif dan berinteraksi
	Message	Pesan antar object, dan menggambarkan urutan kejadian
	Message return	Menyatakan arah kembali antara urutan kejadian



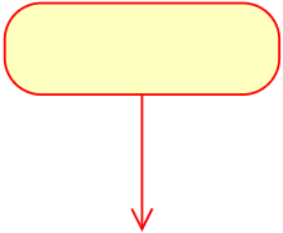
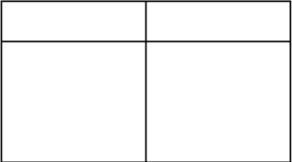
c. Daftar Simbol *Class Diagram*Tabel 2.3 Tabel Daftar Simbol *Class Diagram* (Maharani, 2018)

Simbol	Nama	Keterangan
	Class	<i>Class diagram</i> ini terdiri dari nama kelas, atribut kelas, dan metode / <i>operation</i> (fungsi yang dimiliki suatu kelas)
	Asosiasi	Menyatakan hubungan statis antar <i>class</i> , dan di simbolkan dengan garis tegas saja.
	Agregasi	Hubungan yang menyatakan terdiri atas, dimana <i>class</i> yang satu merupakan bagian dari <i>class</i> lain, namun kedua <i>class</i> ini dapat berdiri sendiri.
	Komposisi	Bentuk khusus dari agragasi dimana <i>class</i> yang menjadi bagian, baru dapat dibuat setelah <i>class</i> yang menjadi <i>whole</i> dibuat.

d. Daftar Simbol Activity Diagram

Tabel 2.4 Tabel Daftar Simbol *Activity Diagram*(Maharani, 2018)

Simbol	Nama	Keterangan
--------	------	------------

	Start Point	Merupakan awal penelusuran. Sebuah activity diagram selalu dimulai dengan start poin
	End Point	Merupakan akhir dari penelusuran. Sebuah activity diagram selalu diakhiri dengan End Point
	Activities	Activity menggambarkan proses, disisi dengan kata kerja atau merupakan state dari sistem yang mencerminkan eksekusi dari suatu aksi.
	Swimlane Style	Sebuah cara untuk mengelompokan activity berdasarkan actor. Actor bisa ditulis dengan nama actor.

2.13 Pengertian Grafik

Grafik ialah sebagai suatu kombinasi antara angka, huruf, simbol, gambar, lambang, perkataan, lukisan, yang disajikan satu media untuk memberi konsep maupun ide dari pengirim kepada sasarannya dalam proses menyampaikan maklumat.

Ada juga pengertian Grafik yang lainnya yaitu merupakan suatu rangka untuk membentuk objek visualisasi tabel. Tabel yang terdiri dari angka-angka dapat disajikan atau dapat ditampilkan ke dalam bentuk gambar, bisa dalam berbentuk garis, lingkaran, batang dll. Dan definisi yang ketiga yaitu gambaran dari pasang surutnya suatu keadaan atau data yang ada dengan garis ataupun gambar. Grafik dibedakan menjadi 3 “tiga” macam, diantaranya yaitu grafik: batang, garis dan lingkaran.

2.14 Mahasiswa

2.14.1 Pengertian Mahasiswa

Menurut Hartaji (2012), mahasiswa adalah seseorang yang tengah menimba ilmu atau belajar dan terdaftar pada salah satu bentuk perguruan tinggi, yang terdiri dari akademi, politeknik, sekolah tinggi, institut, hingga universitas.

Sementara itu, Siswoyo (2007) juga mengemukakan definisi mahasiswa yakni individu yang sedang menuntut ilmu di tingkat perguruan tinggi, baik negeri maupun swasta, atau lembaga lain yang setingkat dengan perguruan tinggi.

Lalu, menurut Daldiyono (2009), mahasiswa adalah seseorang yang sudah lulus dari Sekolah Menengah Akhir (SMA) dan tengah menempuh pendidikan tinggi.

Mahasiswa biasanya dinilai memiliki tingkat intelektualitas yang tinggi, kecerdasan dalam berpikir, serta perencanaannya dalam bertindak. Maka dari itu, berpikir kritis dan bertindak secara cepat serta tepat menjadi sifat yang cenderung melekat pada diri setiap mahasiswa. Seorang mahasiswa biasanya berusia 18

hingga 25 tahun. Pada usia tersebut merupakan masa akhir dari remaja dan masa awal untuk fase dewasa, sehingga dapat disebut bahwa usia mahasiswa adalah fase dimana individu dapat memantapkan pendirian hidupnya.

Berdasarkan pendapat dari beberapa tersebut, dapat disimpulkan bahwa mahasiswa adalah status bagi seseorang yang berusia 18 hingga 25 tahun dan telah lulus dari Sekolah Menengah Atas (SMA) lalu melanjutkan pendidikannya di salah satu bentuk perguruan tinggi.

2.14.2 Tugas dan Kewajiban Mahasiswa

Menurut Siallagan (2011), di lingkungan kampus, mahasiswa akan berperan sebagai masyarakat kampus yang mempunyai tugas utama berupa belajar, membaca buku yang relevan dengan materi perkuliahan, membuat makalah, presentasi, berdiskusi, hadir di sebuah seminar, dan kegiatan lain yang bercorak kekampuasan.

Di samping itu, mahasiswa juga memiliki tugas lain yakni sebagai agen perubahan dan pengontrol sosial masyarakat. Nah, tugas inilah yang nantinya dapat menjadikan seorang mahasiswa sebagai harapan bangsa di masa depan kelak dengan mencari solusi dari berbagai masalah yang tengah dihadapi.

Sama halnya dengan individu lain yang berperan sebagai masyarakat sosial, maka mahasiswa juga memiliki kewajiban untuk menjaga nama baik universitas karena disitulah mereka “tinggal” sekarang.

Berikut adalah beberapa kewajiban yang harus dijalankan mahasiswa,

- a. Bertaqwa dan berakhlak mulia

- b. Mematuhi segala peraturan yang berlaku di, baik di tingkat jurusan, fakultas, hingga universitas.
- c. Terlibat aktif dalam kegiatan kemahasiswaan.
- d. Menjaga nama baik, citra, dan kehormatan universitas.
- e. Saling menghormati sesama mahasiswa dan bersikap sopan kepada dosen hingga karyawan.

2.15 Website

2.15.1 Pengertian Website

Website adalah kumpulan halaman situs yang terdapat dalam sebuah domain atau subdomain pada jaringan *World Wide Web (WWW)* di Internet.

Di era digital saat ini, internet merupakan salah satu sarana yang banyak digunakan untuk memudahkan berbagai kegiatan.

Setiap orang dapat mengakses informasi terbaru dari manapun melalui *website* dengan menggunakan berbagai perangkat mulai *smartphone*, tablet, laptop hingga PC hanya dengan menggunakan koneksi internet.

Website merupakan salah satu *platform* yang paling sering di akses untuk mencari berbagai informasi dan sarana komunikasi. Terlebih saat ini sudah banyak aplikasi dan tutorial cara membuat *website* sendiri dari nol tanpa coding yang membuat proses pembuatan *web* semakin mudah dan pertumbuhan jumlah *website* selalu meningkat di Indonesia.

Penyebaran informasi yang cepat, dan efisien inilah yang menjadi alasan utama mengapa *website* akan selalu menjadi sarana penting untuk mendapatkan dan mengelola informasi.

Pengertian *Website* Menurut Para Ahli

Ada beberapa pengertian *website* menurut para ahli adalah:

1. Gregorius (2000)

Website merupakan kumpulan *web* yang saling terhubung dan seluruh file saling terkait. *Web* terdiri dari halaman dan kumpulan halaman yang disebut dengan *homepage*.

2. Lukmanul Hakim (2004)

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen dalam *website* disebut dengan *webpage* dan link dalam *website* dapat digunakan oleh pengguna untuk beralih dari satu halaman ke halaman (*hyertext*) lain baik antar halaman yang disimpan di server yang sama maupun dalam server yang ada di seluruh dunia. Halaman (*page*) dapat diakses atau dibaca melalui *browser* seperti *Google Chrome*, *Mozilla Firefox* dan lain sebagainya.

3. Azis Sholehul (2013)

Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses diseluruh dunia selama terkoneksi dengan jaringan internet. *Website* juga merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, dan animasi sehingga menarik untuk dikunjungi.

4. Hidayat (2010)

Website adalah kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar, animasi suara atau gabungan dari semuanya baik yang bersifat statis dan dinamis yang membentuk satu rangkaian saling terkait, yang masing-masing dihubungkan dengan jaringan terkait.

5. Yeni Susilowati (2019)

Website adalah sejumlah halaman *web* yang memiliki topik saling terkait antar satu halaman dan halaman yang lainnya, yang biasanya ditempatkan pada sebuah *server web* yang dapat di akses melalui jaringan internet maupun jaringan wilayah lokal (*LAN*).

6. Abdullah (2015)

Website merupakan kumpulan-kumpulan halaman yang terdiri beberapa laman yang didalamnya terdapat informasi digital dalam bentuk gambar, teks, audio, musik dan nimasi lainnya yang disediakan melalui jalur atau koneksi internet.

7. Bekti (2015)

Website adalah kumulan beberapa halaman yang dipakai untuk menampilkan informasi berupa teks, gambar, animasi, suara dan atau bahkan penggabungan dari semuanya.

2.15.2 Jenis – jenis Website

Berdasarkan sifatnya, *website* dapat dibagi menjadi 2, yaitu:

1. Website statis

Website statis adalah *website* yang mempunyai tampilan tetap tidak berubah. Konten dari *website* statis juga sama, jarang sekali terjadi update.

Biasanya suatu *website* statis dibuat menggunakan kode HTML oleh seorang developer. Pembuatnya juga harus memahami kode HTML dan pemrograman. *Website* ini dibuat dengan bantuan text editor atau program design *website* seperti Adobe Dream Weaver. Saat ini untuk membuat *website* statis sudah disediakan sarana sendiri seperti gohugo.io dan hexo.io.

Kelebihan *website* statis adalah cepat dan efisien untuk dikembangkan serta disimpan pada web hosting. Sedangkan kekurangannya adalah membutuhkan skill khusus untuk mengubah konten, kurang bermanfaat karena konten menjadi cepat usang (out of date).

2. Website dinamis

Website dinamis adalah *website* yang kontennya selalu diperbarui secara berkala. Perubahan atau penambahan konten bisa dilakukan oleh seorang developer atau dilakukan oleh beberapa orang yang memiliki akses untuk memperbarui konten. Kebanyakan *website* saat ini berupa *website* dinamis karena pengelolaannya yang mudah.

Keuntungan dari *website* dinamis ini adalah memiliki banyak fitur tambahan, lebih mudah menambah konten baru untuk menambah pengunjung baru dari search engine, dan lebih mudah dikelola bahkan oleh 2 orang atau lebih. Sedangkan kekurangan *website* dinamis adalah membutuhkan waktu dan dana yang lebih banyak untuk melakukan pengembangan serta biaya hosting yang lebih mahal.