

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu ini, penulis menemukan beberapa penelitian sebelumnya yang dapat menjadi referensi penulis dalam mengangkat penelitian yang dilakukan oleh penulis.

Referensi penelitian pertama yaitu penelitian yang ditulis oleh Firman Arifin, dkk. (2007) yang berjudul “*Sistem Monitoring Traffic Paket Internet Melalui SMS berbasis Simple Network Management Protocol*”. Pada penelitian tersebut dibuat aplikasi untuk memantau *traffic* dari paket internet. Hasil data dari *traffic* tersebut ditampilkan melalui *SMS (Short Message Service)* yang berisi informasi paket internet yang masuk dan keluar dalam sebuah jaringan. Dalam membangun aplikasi tersebut, diperlukan protocol *SNMP (Simple Network Management Protocol)* yang digunakan untuk menampilkan hasil data dari pemantauan *traffic* tersebut. Kemudian hasil data tersebut diolah dengan bahasa pemrograman PHP dan disimpan dalam *database*. Aplikasi Monitoring paket internet melalui SMS ini dibangun dengan Bash Script, sebuah *tools scripting* berbasis Unix dan memanfaatkan *SMS Server Tools* sebagai *SMS Gateway*. Dengan sistem *monitoring traffic* paket internet melalui SMS berbasis *SNMP* ini diharapkan suatu jaringan dapat termonitor oleh administrator dari manapun dan kapanpun.

Referensi penelitian kedua yaitu penelitian yang ditulis oleh Ary Mazharuddin Shiddiqi dan Andhika Panji Nugraha (2012) yang berjudul “*Sistem Monitoring Jaringan Dengan Protokol SNMP Menggunakan Piranti Bergerak*”. Pada penelitian tersebut dibuat sebuah aplikasi *mobile* berbasis *Android* dengan bahasa pemrograman

berupa PHP serta protokol SNMP (*Simple Network Management Protocol*). Sistem ini memanfaatkan parameter MIB (*Management Information Base*) untuk mendapatkan kondisi teraktual dari router yang dipantau. Apabila terdapat suatu kondisi yang mengkhawatirkan, sistem akan menginformasikan kepada pengguna/admin. Hasil uji coba aplikasi tersebut menunjukkan bahwa sistem ini dapat menampilkan objek-objek yang dimonitor pada perangkat *mobile* dengan baik. Hal ini ditunjukkan dengan berfungsinya sistem *monitor* dengan baik dan informasi yang ditampilkan adalah akurat.

Referensi penelitian ketiga yaitu penelitian yang ditulis oleh Zaeni Miftah (2019) yaitu “*Penerapan Sistem Monitoring Jaringan Dengan Protokol SNMP Pada Router Mikrotik dan Aplikasi Dude Studi Kasus Stikom CKP*”. Pada penelitian ini berfokus pada penerapan Protokol SNMP pada sebuah jaringan komputer sehingga menghasilkan suatu sistem *monitoring* jaringan yang dapat mempermudah administrator jaringan untuk memantau jaringan secara *real-time*. Sistem ini juga memanfaatkan parameter MIB (*Management Information Base*) untuk menampilkan kondisi jaringan secara *real-time*. Hasil dari Parameter tersebut akan dipantau secara *real-time* menggunakan program *The Dude*, yaitu program NMS (*Network Monitoring and Management System*) yang disediakan oleh Mikrotik secara gratis. *The Dude* didesain untuk menampilkan semua perangkat yang terhubung jaringan dalam bentuk map atau topologi. Apabila suatu perangkat dapat terhubung ke internet, maka tampilan indikator di program *The Dude* akan menunjukkan warna hijau. Sedangkan jika suatu perangkat terputus koneksi internetnya, maka tampilan indikator di program *The Dude* akan menunjukkan warna merah.

Referensi penelitian keempat yaitu penelitian yang ditulis oleh Susmini Indriani Lestaringati dan Fathur Rozak (2014) yang berjudul “*Pembangunan Aplikasi Monitoring Jaringan Berbasis Web Menggunakan Simple Network Management Protocol*”. Pada penelitian ini berfokus pada pembuatan aplikasi *monitoring* jaringan berbasis web yang menggunakan protokol SNMP dalam

penerapannya. Pada aplikasi *monitoring* jaringan berbasis web ini dilengkapi dengan fitur *login* untuk admin dengan memasukkan *username* dan *password*. Adapun fitur utama yang terdapat pada aplikasi *monitoring* jaringan berbasis *web* ini yaitu fitur untuk menambah *agent* atau *device* yang ingin dipantau jaringannya yang berisi 4 parameter yaitu *Host Name*, *Community Name*, *IP Address*, dan *Type of Device* (Komputer atau *Router*). Adapun hasil dari *Agent* yang sudah diinputkan di aplikasi dapat kita pantau melalui fitur *Device List* yang menampilkan kondisi jaringan dari *agent* atau perangkat yang sudah kita masukkan.

Referensi penelitian terakhir yaitu penelitian yang ditulis oleh Lukas Tri Christanto (2016) yang berjudul “*Perancangan Sistem Monitoring Jaringan Melalui Perangkat Android Berbasis Simple Network Management Protocol*”. Pada penelitian ini berfokus pada perancangan sistem *monitoring* jaringan berbasis *Android* dengan memanfaatkan protokol SNMP untuk menampilkan kondisi jaringan secara *real-time*. Sistem ini dilengkapi dengan fitur *login* untuk masuk ke dalam sistem tersebut. Adapun fitur yang tersedia pada sistem ini yaitu fitur *Server Info* yaitu Informasi mengenai *server* yang tersambung, *Network Info* yaitu informasi mengenai Jaringan yang tersambung, dan juga *Resources* yaitu informasi mengenai perangkat jaringan yang tersambung. Tentunya semua fitur ini dapat dipantau oleh administrator jaringan secara *real-time* dan akurat.

## **2.2 Aplikasi**

Menurut Indrajani (2018:3), aplikasi adalah program yang menentukan aktifitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer. Setiawan dan Ramdany (2019:9), berpendapat bahwa aplikasi adalah komponen yang berguna untuk melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data.

Sedangkan menurut *Kamus Besar Bahasa Indonesia*, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau

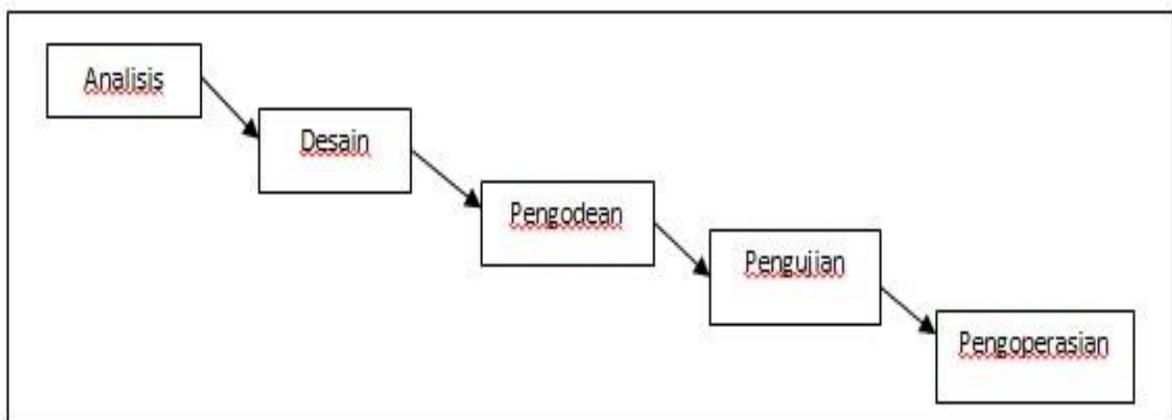
ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

Jadi, dapat disimpulkan bahwa aplikasi adalah penerapan dari rancangan sistem berupa program siap pakai yang dapat digunakan untuk menjalankan perintah ataupun mengolah data yang bertujuan untuk menjalankan dan melaksanakan tugas khusus dari pengguna (*user*).

### 2.3 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan penulis pada laporan ini yaitu model SDLC air terjun (*waterfall*). Menurut Rosa dan Shalahuddin (2019:28), model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pengoperasian. Model air terjun pada dasarnya mengikuti prinsip dari sebuah air terjun. Dimana setiap pekerjaan akan dilakukan secara berurutan mulai dari atas hingga ke bawah.

Berikut adalah skema model SDLC air terjun (*waterfall*) :



**Gambar 2.1 Ilustrasi Model *Waterfall***

Adapun uraian atau penjelasan dari setiap tahapan yang terjadi pada model air terjun yaitu sebagai berikut :

### **1. Analisis**

Pada tahap awal, kita menganalisis atau mempersiapkan kebutuhan dari rancangan aplikasi yang akan dibuat. Informasi dan *insight* yang diperoleh dapat berupa dari hasil wawancara, survei, studi literatur, observasi, hingga diskusi. Dan untuk penelitian ini, penulis melakukan analisis informasi dari studi literatur berupa bahan bacaan seperti jurnal ataupun laporan akhir terdahulu, dan juga studi lapangan, yaitu mengamati kondisi yang terjadi pada lapangan. Dalam hal ini dilakukan di Jurusan Teknik Komputer Politeknik Negeri Sriwijaya.

### **2. Desain**

Tahap kedua yaitu pembuatan desain aplikasi sebelum masuk pada tahap pengodean (*coding*). Tujuan dari tahap ini, supaya mempunyai gambaran jelas mengenai tampilan dan antarmuka aplikasi yang kemudian akan dieksekusi di tahap selanjutnya. Untuk proses ini, penulis berfokus pada pembangunan struktur data, arsitektur aplikasi, perancangan antarmuka, konfigurasi jaringan yang dipakai, hingga perancangan fungsi internal dan eksternal dari setiap algoritma prosedural.

### **3. Pengodean (*Coding*)**

Tahapan berikutnya adalah implementasi kode program dengan menggunakan berbagai *tools* dan bahasa pemrograman sesuai dengan kebutuhan. dimana hasil dari desain aplikasi pada tahapan sebelumnya akan diterjemahkan ke dalam bahasa pemrograman. Selain itu, pada tahap ini juga dilakukan pemeriksaan lebih dalam terkait dengan aplikasi yang sudah dibuat, apakah berjalan dengan semestinya atau tidak.

#### 4. Pengujian

Tahap yang keempat, masuk dalam proses pengujian sistem. Pada tahap ini, akan dilakukan pengujian aplikasi yang sudah diprogram pada tahap sebelumnya. Tujuan dari tahap pengujian ini ialah untuk mengetahui apakah aplikasi sudah sesuai dengan desain, dan fungsionalitas dari aplikasi tersebut apakah berjalan dengan baik atau tidak. Jadi, dengan adanya tahap pengujian, maka dapat mencegah terjadinya kesalahan, *bug*, atau *error* pada program sebelum masuk pada tahap selanjutnya.

#### 5. Pengoperasian

Tahap terakhir pada model SDLC air terjun (*waterfall*) ini adalah pengoperasian dari aplikasi. Setelah dilakukan pengujian sistem, maka akan masuk pada tahap produk dan pemakaian perangkat lunak oleh pengguna (*user*). Untuk proses pemeliharaan, memungkinkan pengembang untuk melakukan perbaikan terhadap kesalahan yang ditemukan pada aplikasi setelah digunakan oleh *user*.

### 2.4 Monitoring

Menurut Peraturan Pemerintah Nomor 39 Tahun 2006, *monitoring* merupakan suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu, dengan tujuan agar semua data masukan atau informasi yang diperoleh dari hasil pengamatan tersebut dapat menjadi landasan dalam mengambil keputusan tindakan selanjutnya yang diperlukan.

Dalam konteks jaringan komputer, kegiatan *monitoring* yaitu mengamati suatu keadaan atau kondisi jaringan komputer (dalam hal ini bisa berupa kondisi kualitas jaringan ataupun kondisi perangkat jaringan), untuk memperoleh informasi atau data masukan yang dapat menjadi landasan untuk mengambil tindakan lebih lanjut seperti perawatan dan perbaikan jaringan komputer.

*Monitoring* sendiri memiliki berbagai fungsi. Menurut William E. Dunn (2017), monitoring mempunyai empat fungsi utama, yaitu :

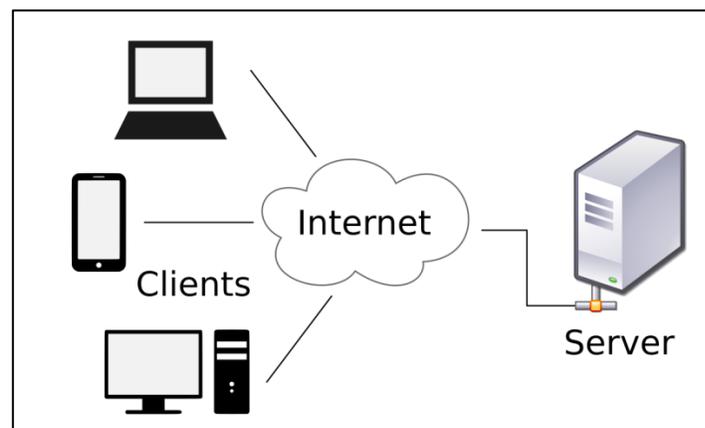
1. **Ketaatan (*Compliance*)**. *Monitoring* menentukan apakah tindakan administrator, staf, dan semua yang terlibat mengikuti standar dan prosedur yang telah disiapkan.
2. **Pemeriksaan (*Auditing*)**. *Monitoring* menetapkan apakah sumber dan layanan yang diperuntukkan bagi pihak tertentu (*target*) telah mencapai mereka.
3. **Laporan (*Accounting*)**. *Monitoring* menghasilkan informasi yang mampu menghitung hasil perubahan sosial dan masyarakat sebagai akibat implementasi kebijaksanaan sesudah periode waktu tertentu.
4. **Penjelasan (*Explanation*)**. *Monitoring* menghasilkan informasi yang membantu menjelaskan bagaimana akibat kebijaksanaan dan mengapa antara perencanaan dan pelaksanaannya tidak cocok.

## 2.5 Jaringan Komputer

Menurut Andrew S. Tanenbaum dan David J. Weatherhall (2011), jaringan komputer merupakan suatu sistem yang memungkinkan beberapa komputer untuk saling terhubung satu sama lain. Perangkat seperti komputer merupakan komponen utama dari sistem ini yang biasanya digunakan oleh pengguna (*user*) untuk dapat mengirimkan dan/atau menerima informasi dari orang lain sebagai pengguna pada komputer yang berbeda.

Setiap bagian dari jaringan komputer dapat meminta dan memberikan layanan (*service*). Pihak yang meminta/menerima layanan disebut klien (*client*) dan yang memberikan/mengirim layanan disebut peladen (*server*). Desain sistem yang terbentuk antara penerima dan pengirim layanan disebut dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi dari jaringan komputer. Desain sistem *client-server* memiliki beberapa keuntungan diantaranya :

1. Dapat menyimpan *big data* atau data yang berukuran besar, melalui desain sistem *client-server* ini *client* dapat menyimpan dan mengakses *database* yang sangat besar.
2. Menaikkan kinerja, dengan konsep *client server*, pengguna dapat meningkatkan kinerja akses data berupa informasi. Kinerja cepat dikarenakan seluruh data disimpan pada satu *server* yang *multi tasking*.
3. Mengurangi biaya komunikasi antar pengguna.
4. Konsisten, melalui konsep ini *database* yang diberikan akan meningkatkan konsistensi data. Karena satu data dapat diakses oleh beberapa *user* secara bersamaan.



**Gambar 2.2** Arsitektur *Client-Server*

Dalam mempelajari jaringan komputer, tentunya kita akan dihadapkan dengan beberapa jenis jenis dari jaringan komputer itu sendiri. Secara teori, klasifikasi dari jaringan komputer dapat dibagi menjadi 2, yaitu berdasarkan transmisi dan geografis.

### 2.5.1 Jaringan Komputer Berdasarkan Transmisi

Secara teknologi transmisi, jaringan komputer dibagi menjadi dua jenis, yaitu jaringan *broadcast* dan jaringan *point-to-point*.

1. Jaringan **Broadcast** memiliki saluran komunikasi tunggal yang dipakai bersama-sama oleh semua perangkat yang terkoneksi ke jaringan. Pesan-

pesan berukuran kecil yang dikirimkan oleh suatu mesin akan diterima oleh mesin-mesin lainnya. *Field* alamat pada sebuah paket berisi keterangan tentang kepada siapa paket tersebut ditujukan. Saat menerima paket, mesin akan mencek *field* alamat. Bila paket tersebut ditujukan untuk dirinya, maka mesin akan memproses paket itu, bila paket ditujukan untuk mesin lainnya, mesin tersebut akan mengabaikannya.

2. Jaringan ***Point-to-Point*** terdiri dari beberapa koneksi pasangan individu, dari satu perangkat ke perangkat lain. Untuk mengirim paket dari sumber ke suatu tujuan, sebuah paket pada jaringan jenis ini mungkin harus melalui satu atau lebih mesin-mesin perantara. Seringkali harus melalui banyak rute yang mungkin berbeda jaraknya. Karena itu algoritma *route* memegang peranan penting pada jaringan *point-to-point*.

### 2.5.2 Jaringan Komputer Berdasarkan Geografis

Secara cakupan geografis, jaringan komputer dibagi menjadi empat jenis, yaitu jaringan LAN, MAN, WAN, dan Internet.

#### 1. ***Local Area Network (LAN)***

*Local Area Network (LAN)* dapat didefinisikan sebagai kumpulan komputer yang saling dihubungkan bersama didalam satu area tertentu yang tidak begitu luas, seperti di dalam satu kantor atau gedung. LAN dapat juga didefinisikan berdasarkan pada penggunaan alamat IP komputer pada jaringan. Suatu komputer atau host dapat dikatakan satu LAN bila memiliki alamat IP yang masih dalam satu alamat jaringan, sehingga tidak memerlukan router untuk berkomunikasi.

#### 2. ***Metropolitan Area Network (MAN)***

*Metropolitan Area Network (MAN)* pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya memakai teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang berdekatan dan dapat dimanfaatkan untuk keperluan pribadi atau umum.

### 3. *Wide Area Network (WAN)*

*Wide Area Network (WAN)* merupakan jaringan komputer yang mencakup daerah geografis yang luas, sering kali mencakup sebuah negara atau benua.

### 4. **Internet**

Internet (*interconnection-networking*) adalah seluruh jaringan komputer yang saling terhubung menggunakan standar sistem global *Transmission Control Protocol/Internet Protocol Suite (TCP/IP)* sebagai protokol pertukaran paket untuk melayani miliaran pengguna di seluruh dunia, bahkan antar planet.

## 2.5.3 Perangkat Jaringan

Dalam menyusun sebuah jaringan komputer, peran perangkat jaringan komputer sangatlah penting. Perangkat jaringan adalah semua komputer, *peripheral*, *interface card*, dan perangkat tambahan yang terhubung ke dalam suatu sistem jaringan komputer untuk melakukan komunikasi data. Adapun perangkat yang umum dipakai pada pembangunan jaringan komputer terdiri dari :

### 1. *Server*

*Server* merupakan pusat kontrol dari jaringan komputer. *Server* berfungsi untuk menyimpan informasi dan untuk mengelola suatu jaringan komputer. *Server* akan melayani seluruh *client* atau *workstation* yang terhubung ke jaringan. Sistem operasi yang digunakan pada *server* adalah sistem operasi yang khusus yang dapat memberikan layanan bagi *workstation*.



**Gambar 2.3 Server**

## 2. *Workstation*

*Workstation* adalah komputer beserta perangkat pendukung yang terhubung dengan sebuah LAN. Semua komputer yang terhubung dengan jaringan dapat dikatakan sebagai *workstation*. Komputer ini yang melakukan akses ke *server* guna mendapat layanan yang telah disediakan oleh *server*.



**Gambar 2.4 Workstation (Komputer Desktop)**

## 3. *Router*

*Router* adalah perangkat yang berfungsi menghubungkan suatu LAN ke suatu *internetworking* dan mengelola penyaluran lalu-lintas data di dalamnya. *Router* akan menentukan jalur terbaik untuk komunikasi

data. *Router* bekerja pada layer *network* dari model OSI untuk memindahkan paket-paket antar jaringan menggunakan alamat logikanya. *Router* memiliki tabel *routing* yang melakukan pencatatan terhadap semua alamat jaringan yang diketahui dan lintasan yang mungkin dilalui serta waktu tempuhnya. Setiap router mempunyai fasilitas *DHCP* (*Dynamic Host Configuration Procotol*) dan juga *NAT* (*Network Address Translator*).



**Gambar 2.5 Router**

#### **4. Switch**

*Switch* adalah perangkat jaringan komputer yang berfungsi untuk menghubungkan *multiple* komputer. *Switch* juga bekerja di lapisan *data link*. Cara kerja *Switch* ini mirip seperti *bridge*, namun *Switch* mempunyai beberapa port yang sering disebut dengan *multi-port bridge*. Fungsi *Switch* adalah untuk manajemen lalu lintas yang ada pada jaringan komputer. Alat ini mempunyai tugas untuk mentransfer suatu paket data hingga sampai ke tujuan dengan perangkat yang tepat. *Switch* ini juga mempunyai fungsi untuk mencari jalur yang sangat baik dan optimal serta memastikan pengiriman paket data yang efisien pada tujuannya.



**Gambar 2.6 Switch**

### 5. *Bridge*

Sesuai dengan namanya, *Bridge* adalah perangkat yang berfungsi untuk meneruskan lalu lintas antara segmen jaringan berdasarkan informasi pada lapisan *data link*. Segmen ini mempunyai alamat lapisan jaringan yang sama. *Bridge* bekerja dengan mengenali alamat MAC asal yang mentransmisi data ke jaringan dan secara otomatis membangun sebuah tabel internal. Tabel ini berfungsi untuk menentukan ke segmen mana paket akan di *route* dan menyediakan kemampuan *filtering*. *Bridge* membagi satu buah jaringan besar kedalam beberapa jaringan kecil. Bridge juga dapat di gunakan untuk mengkoneksi *network* yang menggunakan tipe kabel yang berbeda ataupun topologi yang berbeda pula.



**Gambar 2.7 Bridge**

## 6. Kabel Jaringan

Kabel jaringan adalah perangkat keras yang memiliki bentuk kabel memanjang dan memiliki kegunaan sebagai koneksi jaringan. Kabel jaringan digunakan untuk menghubungkan antar komputer atau lebih dengan berbagai topologi jaringan yang digunakan atau sebagai transmisi antara *server* kepada *client* dan juga sebaliknya. Kabel jaringan secara fungsi hanya untuk koneksi jaringan saja, tidak bisa untuk kegunaan lainnya seperti contoh sebagai kabel untuk aliran listrik. Kabel jaringan memiliki beragam jenis yang masing masing disusun dengan struktur dan kegunaan yang berbeda seperti kabel *fiber optic* yang sekarang ini banyak dipergunakan karena kecepatan transfer datanya.



**Gambar 2.8 Kabel Jaringan (*twisted pair*)**

### 2.6 Mobile

Kata *Mobile* berasal dari Bahasa Inggris yang artinya mudah berubah atau mudah dipindahkan. Dalam konteks teknologi, *Mobile* adalah sebuah sistem perangkat lunak yang memungkinkan setiap pemakai melakukan mobilitas dengan perlengkapan *PDA*-asisten digital perusahaan pada telepon genggam atau seluler (Wongso, 2018). Jadi, dapat disimpulkan bahwa *Mobile* merupakan suatu sistem yang membuat *user* dapat mengakses dan mengontrol sistem tersebut di mana saja secara fleksibel, tanpa harus berdiam diri di suatu tempat, dan terjadi pemutusan atau

terputusnya komunikasi. Adapun contoh dari perangkat *mobile* yang sangat populer adalah telepon seluler.

## **2.7 Android**

### **2.7.1 Pengertian Android**

Android merupakan sebuah sistem operasi telepon seluler dan komputer tablet layar sentuh (*touch screen*) yang berbasis Linux. Platform Android terdiri dari sistem operasi Linux, sebuah GUI (*Graphic User Interface*), sebuah *Web Browser*, dan Aplikasi *End-User* yang dapat diunduh dan juga para pengembang bisa dengan leluasa berkarya dan menciptakan aplikasi yang terbaik serta terbuka untuk digunakan bagi semua perangkat.



**Gambar 2.9 Logo Android**

### **2.7.2 Versi-Versi OS Android**

Sejak pertama kali diperkenalkan secara umum pada tahun 2007, Android telah mengalami perkembangan yang begitu pesat, ditandai dengan banyaknya versi sistem operasi Android ini. Berikut adalah versi versi Android yang sudah diperkenalkan sampai sekarang beserta tanggal rilisnya :

**Tabel 2.1 Versi-versi OS Android**

<b>No.</b>	<b>Versi Android</b>	<b>Tanggal Rilis</b>
1	Android 1.0 Alpha	23 September 2008
2	Android 1.1 Beta	9 Februari 2009
3	Android 1.5 Cupcake	30 April 2009
4	Android 1.6 Donut	15 September 2009
5	Android 2.0 Eclair	26 Oktober 2009
6	Android 2.2 Froyo	20 Mei 2010
7	Android 2.3 Gingerbread	6 Desember 2010
8	Android 3.0 Honeycomb	22 Februari 2011
9	Android 4.0 Ice Cream Sandwich	19 Oktober 2011
10	Android 4.1 Jelly Bean	27 Juni 2012
11	Android 4.4 KitKat	31 Oktober 2013
12	Android 5.0 Lollipop	25 Juni 2014
13	Android 6.0 Marshmallow	5 Mei 2015
14	Android 7.0 Nougat	19 Oktober 2016
15	Android 8.0 Oreo	21 Agustus 2017

16	Android 9.0 Pie	6 Agustus 2018
17	Android 10	3 September 2019

### 2.7.3 Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas *.NET Compact Framework*).



**Gambar 2.10 Logo Visual Studio**

## 2.7.4 Xamarin

Xamarin adalah *platform* pengembangan aplikasi *mobile* dari Microsoft. Dengan Xamarin anda dapat membuat aplikasi *mobile cross platform* pada perangkat Android, IOS, dan Windows 10 (*Universal Windows Platform*). Saat ini terdapat 3 produk Xamarin yang dapat anda pilih yaitu:

- Xamarin for Android
- Xamarin for IOS
- Xamarin Forms

Seperti namanya Xamarin for Android digunakan jika anda ingin membuat aplikasi *native* Android. Sedangkan Xamarin for IOS digunakan untuk membuat aplikasi *native* IOS. Jika anda menggunakan salah satu atau kedua teknologi tersebut maka anda dapat membagi komponen yang sudah anda buat pada satu proyek untuk digunakan di proyek yang lain. Namun yang dapat dibagi hanya komponen selain UI (*User Interface*). Xamarin for Android dan Xamarin for IOS masih menggunakan komponen UI yang spesifik dengan platform.

*Xamarin Forms* adalah teknologi yang lebih baru, kelebihan dari *Xamarin Forms* adalah dapat menggunakan satu UI yang dapat di gunakan pada platform yang berbeda. Dengan *Xamarin Forms* anda cukup mengembangkan satu aplikasi dengan target platform yang berbeda. *Xamarin Forms* menggunakan format XAML (*eXtensible Application Markup Language*) untuk membuat tampilan UI.

## 2.8 SNMP (Simple Network Management Protocol)

### 2.8.1 Pengertian SNMP

*Simple Network Management Protocol* atau disingkat SNMP merupakan protokol untuk manajemen dan pemantauan kondisi peralatan yang terhubung dalam jaringan IP (*Internet Protocol*). Peralatan-peralatan itu antara lain *switch*, *router*, modem, komputer, server dan lain-lain. SNMP menggunakan data-data yang didapatkan dari komunikasi UDP dengan perangkat yang masuk dalam jaringan

tersebut. SNMP dapat meminta data ataupun melakukan setting kepada peralatan yang bersangkutan.

Sistem manajemen jaringan dengan SNMP mempunyai dua entitas utama yaitu Manajer dan agen. Manajer adalah tempat di mana administrator jaringan mengontrol fungsi manajemen jaringan. Sedangkan agen adalah entitas yang melekat ke perangkat yang sebenarnya dikelola. *Switch, Router* atau *Server* jaringan adalah contoh perangkat yang dikelola. SNMP disebut sebagai “*simple*” karena agen membutuhkan *software* yang sangat kecil. Pemrosesan dan penyimpanan data berada pada sistem Manajer, sementara bagian pelengkap dari fungsi-fungsi berada di sistem yang dikelola (dikenal sebagai obyek). SNMP juga menggunakan datagram UDP karena lebih sederhana dalam pengiriman informasi.

### 2.8.2 Versi-versi SNMP

Saat ini, sudah terdapat 3 versi dari SNMP, yaitu:

#### 1. SNMP versi 1 (SNMPv1)

Merupakan SNMP implementasi paling awal yang beroperasi di atas protokol lain seperti *User Datagram Protokol (UDP)*, *Internet Protokol (IP)*, *OSI Connectionless Network Service (CLNS)*, *AppleTalk Datagram Delivery Protocol* dan *Novell Internet Packet Exchange (IPX)*.

#### 2. SNMP versi 2 (SNMPv2)

Pada versi 2 terdapat peningkatan untuk mendukung penanganan kesalahan yang lebih efisien yang dijelaskan dalam RFC 1901. Pertama kali dikenalkan pada RFC 1441 dan sering disebut sebagai SNMPv2c.

#### 3. SNMP versi 3 (SNMPv3)

Pada versi ini terdapat peningkatan dari sisi keamanan dan privasi yang telah tertulis di RFC 3410.

SNMP versi 2 adalah versi protokol SNMP yang paling umum digunakan saat ini. Pada SNMPv3 menyertakan fitur keamanan baru yang menambahkan dukungan otentikasi dan enkripsi pesan SNMP serta melindungi paket selama transit.

### 2.8.3 *Management Information Base (MIB)*

MIB Adalah struktur basis data variabel dari elemen jaringan yang dikelola. Pada kelompok *interface* terdapat variabel objek MIB yang mendefinisikan karakteristik *interface* diantaranya :

1. **ifInOctets** mendefinisikan jumlah total *byte* yang diterima,
2. **ifOutOctets** mendefinisikan jumlah total *byte* yang dikirim,
3. **ifInErrors** mendefinisikan jumlah paket diterima yang dibuang karena rusak,
4. **ifOutErrors** mendefinisikan jumlah paket dikirim yang dibuang karena rusak, dan variabel objek lainnya yang juga berkaitan dengan paket internet.

### 2.8.4 *Trap*

*Trap* Adalah jenis PDU (*Protocol Data Unit*) yang dikirimkan dari agent kepada manager. SNMP *trap* ini adalah sebuah pesan yang diprakarsai oleh suatu elemen dalam jaringan (*agent*) dan dikirimkan kepada manager untuk memberitahu atau memberikan informasi kepada manager bahwa terjadi suatu event tertentu pada objek yang di-manage (*managed device*).

### 2.8.5 *Object Identifier (OID)*

*Object Identifier (OID)* merupakan sebuah pengenal yang digunakan untuk menamakan sebuah objek yang terdapat dalam MIB. OID bersifat unik untuk masing-masing objek. Secara struktural, sebuah OID terdiri dari sebuah *node* dalam *namespace* yang ditetapkan secara hirarki, yang didefinisikan secara formal menggunakan standar ASN.1.

## 2.9 **Bahasa Pemrograman**

### 2.9.1 **C#**

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah

dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti *Java*, *Delphi*, *Visual Basic*, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar *ECMA-334 C# Language Specification*, nama *C#* terdiri atas sebuah huruf Latin *C* (U+0043) yang diikuti oleh tanda pagar yang menandakan angka *#* (U+0023). Tanda pagar *#* yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam papan tombol standar. Selain itu *C#* juga bahasa pemrograman yang berorientasi objek, jadi *C#* juga mengukung konsep objek seperti *inheritance*, *class*, *polymorphism* dan *encapsulation*. bahasa *C#* juga bersifat *general-purpose*. Adapun maksud bahasa *C#* bersifat *General-purpose* yaitu bahasa *C#* mampu digunakan untuk menulis *software* komputer secara umum, seperti halaman web ataupun aplikasi yang berbasis *mobile*.



**Gambar 2.11 Logo C#**

### **2.9.2 XAML**

*Extensible Application Markup Language* (XAML) adalah sebuah bahasa berbasis XML deklaratif yang diciptakan oleh Microsoft yang digunakan untuk menginisialisasi nilai dan objek terstruktur. XAML tersedia di bawah *Open Specification Promise Microsoft*. Akronim ini pada mulanya merupakan singkatan

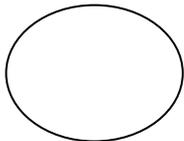
dari *Extensible Avalon Markup Language* - Avalon, yaitu sebuah nama kode untuk *Windows Presentation Foundation* (WPF).

### 2.10 *Data Flow Diagram* (DFD)

Menurut Kadir (2021:23), *Data Flow Diagram* (DFD), untuk menggambarkan proses-proses bisnis dalam organisasi dan sekaligus menerangkan kaitan antara proses dan data, teknik seperti diagram aliran data. Sedangkan Kristianto (2018:61), menyatakan DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Dan menurut Rosa dan Shalahudin (2019:70), *Data Flow Diagram* (DFD) atau dalam Bahasa Indonesia menjadi *Diagram Alir Data* (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang di aplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

**Tabel 2.2. Simbol-simbol *Data Flow Diagram* (DFD)**

NO	SIMBOL	KETERANGAN
	<p style="text-align: center;"><b>Proses</b></p> 	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan di implementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur didalam kode program.</p> <p><b>Catatan:</b> Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>

	<p style="text-align: center;"><b>Storage</b></p> <hr style="width: 50%; margin: auto;"/> <hr style="width: 50%; margin: auto;"/>	<p>File atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi table-table basis data yang dibutuhkan, table-table ini juga harus sesuai dengan perancangan table-table basis data <i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)</p> <p><b>Catatan:</b> Nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>
	<p style="text-align: center;"><b>Entitas Luar</b></p> <div style="text-align: center; border: 1px solid black; width: 100px; height: 40px; margin: auto;"></div>	<p>Entitas Luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p><b>Catatan:</b> Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
	<p style="text-align: center;"><b>Aliran Data</b></p> <div style="text-align: center; margin-top: 20px;">  </div>	<p>Aliran Data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses masukan (<i>input</i>) atau keluaran (<i>output</i>).</p> <p><b>Catatan:</b> Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.</p>

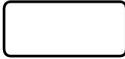
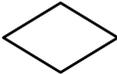
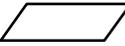
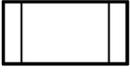
Jadi dapat disimpulkan Data Flow Diagram adalah suatu model logika data yang menggunakan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*) dengan menggunakan notasi-notasi dalam bentuk simbol-simbol.

## 2.11 Flowchart

Menurut I Gusti Ngurah Suryantara (2009), badan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. *Flowchart* digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

Tujuan utama penggunaan *flowchart* adalah untuk menggambarkan suatu tahap penyelesaian masalah secara sederhana, terurai, rapi, dan jelas dengan menggunakan simbol-simbol yang standar. Dalam penulisan *flowchart* dikenal dua model yaitu *flowchart* sistem dan *flowchart* program. *Flowchart* sistem menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antara peralatan tersebut. *Flowchart* program menggambarkan suatu logika dari suatu prosedur pemecahan masalah. Berikut merupakan beberapa simbol pada *flowchart* :

**Tabel 2.3. Simbol pada Flowchart**

No.	Simbol	Nama Simbol	Keterangan
1.		<i>Alternate Process</i>	Menyatakan segala jenis operasi yang diproses dengan menggunakan mesin yang memiliki keyboard
2.		<i>Decision</i>	suatu penyelesaian kondisi dalam program
3.		<i>Data</i>	Mewakili data <i>input</i> atau <i>output</i>
4.		<i>Predefined Process</i>	Suatu operasi yang rinciannya di tunjukkan di tempat lain
5.		<i>Document</i>	Document <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer

6.		<i>Terminator</i>	Untuk menunjukkan awal dan akhir dari suatu proses
7.		<i>Process</i>	Kegiatan proses dari operasi program komputer