BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Teori merupakan salah satu unsur terpenting dalam penelitian yang memiliki peran sangat besar dalam penelitian. Suatu landasan teori dari suatu penelitian disini bisa disimpulkan sebagai studi literatur atau tinjauan pustaka. Hasil dari landasan teori atau kajian teori ini diperoleh kesimpulan-kesimpulan atau pendapat-pendapat para ahli, kemudian dirumuskan pada pendapat baru.

2.2.1 Internet

Internet atau singkatan dari *Inter-Network* adalah sekumpulan jaringan komputer yang saling terhubung dengan menggunakan protokol pertukaran paket (*packet switching communication protocol*) seperti *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP). Internet menyediakan layanan telekomunikasi dan jutaan sumber daya informasi yang dapat diakses di seluruh dunia [1].

Jaringan yang membentuk internet bekerja berdasarkan suatu set protokol standar yang digunakan untuk menghubungkan jaringan komputer dan mengalamati lalu lintas dalam jaringan. Protokol ini mengatur format data yang diijinkan, penanganan kesalahan (*error handling*), lalu lintas pesan, dan standar komunikasi lainnya. Protokol standar pada internet dikenal sebagai TCP/IP *(Transmission Control Protocol/Internet Protocol*). Protokol ini memiliki kemampuan untuk bekerja diatas segala jenis komputer, tanpa terpengaruh oleh perbedaan perangkat keras maupun sistem operasi yang digunakan. Sebuah sistem komputer yang terhubung secara langsung ke jaringan memiliki nama domain dan alamat IP (*Internet Protocol*) dalam bentuk numerik dengan format tertentu sebagai pengenal.

2.2.2 *Internet of Things* (IoT)

Internet of Things (IoT) merupakan teknologi yang memungkinkan adanya pengendalian, komunikasi, dan kerjasama antara berbagai jenis perangkat keras melalui internet, IoT muncul sebagai bentuk perubahan dan perkembangan teknologi informasi dan jaringan. IoT bukan hanya terkait pengendalian jarak jauh, tapi IoT juga berkaitan dengan bagaimana proses untuk berbagi data, memvirtualisasi segala hal nyata kedalam bentuk internet dan lain-lain [1] .

"*A Things*" pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan orang dengan monitor implant jantung, hewan peternakan dengan *transponder biochip*, sebuah mobil

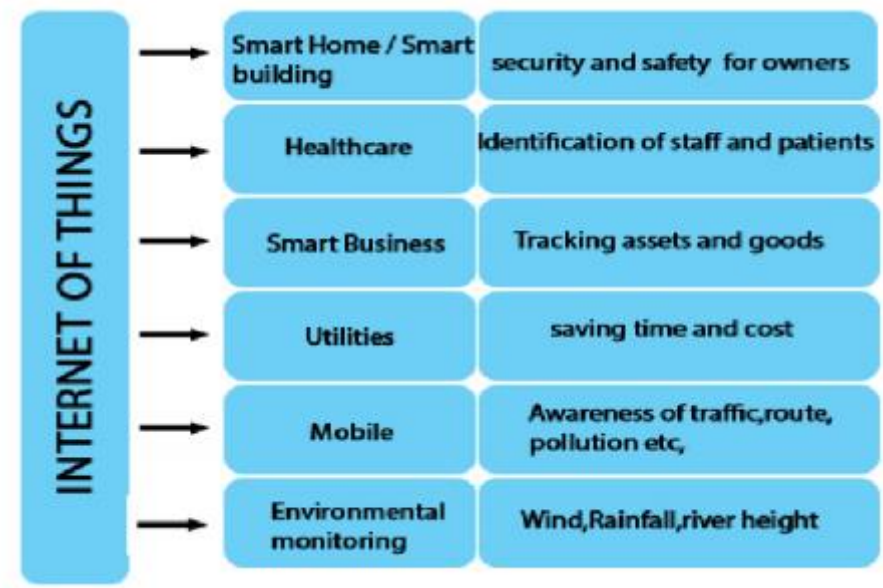
yang telah dilengkapi *built-in* sensor untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, IoT paling erat hubungannya dengan komunikasi *machine-to-machine* (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau "*smart*". (contoh: *smart label*, *smart meter*, smart *grid* sensor) [2].

Meskipun konsep ini kurang populer hingga tahun 1999, namun IoT telah dikembangkan selama beberapa dekade. Alat Internet pertama, misalnya, adalah mesin Coke di Carnegie Melon University di awal 1980-an [3]. Para programmer dapat terhubung ke mesin melalui internet, memeriksa status mesin dan menentukan apakah ada atau tidak minuman dingin yang menunggu mereka, tanpa harus pergi ke mesin tersebut [1].

Istilah IoT (*Internet of Things*) mulai dikenal tahun 1999 yang saat itu disebutkan pertama kalinya dalam sebuah presentasi oleh Kevin Ashton, co-founder and executive director of the Auto-ID Center di MIT [4].

Dengan semakin berkembangnya infrastruktur internet, maka kita menuju babak berikutnya, di mana bukan hanya smartphone atau komputer saja yang dapat terkoneksi dengan internet. Namun berbagai macam benda nyata akan terkoneksi dengan internet. Sebagai contohnya dapat berupa : mesin produksi, mobil, peralatan elektronik, peralatan yang dapat dikenakan manusia (*wearables*), dan termasuk benda nyata apa saja yang semuanya tersambung ke jaringan lokal dan global menggunakan sensor dan atau aktuator yang tertanam [5].

Pengaplikasian dari IoT bisa diklasifikasikan menjadi berbagai macam kegunaan yang bisa dilihat di Gambar 2.6 Contoh Pengaplikasian IoT.



Gambar 2.1 Contoh Pengaplikasian IoT

Teknologi *internet of things* sangat luar biasa. Jika sudah direalisasikan, teknologi ini tentu akan sangat memudahkan pekerjaan manusia. Manusia tidak akan perlu lagi mengatur mesin saat menggunakannya, tetapi mesin tersebut akan dapat mengatur dirinya sendiri dan berinteraksi dengan mesin lain yang dapat berkolaborasi dengannya. Hal ini membuat mesin-mesin tersebut dapat bekerja sendiri dan manusia dapat menikmati hasil kerja mesin-mesin tersebut tanpa harus repot-repot mengatur mereka [5].

Kevin Ashton seorang pelopor teknologi yang juga membuat sistem standar global untuk RFID dan sensor lainnya mengatakan bahwa hampir semua data yang beredar di internet berasal dari hasil input atau hasil *capture* yang dilakukan oleh manusia ke dalam sistem. Dari sudut pandang sistem, manusia adalah obyek yang lambat, rawan kesalahan, pengantar data yang tidak efisien dan memiliki batasan dalam hal kualitas dan kuantitas, bahkan kadang mencoba menterjemahkan dan mengubah data tersebut. Sebagai alternatif akan lebih efisien jika sistem dapat terkoneksi dengan sensor yang dapat menterjemahkan kejadian di dunia nyata secara langsung. Jadi, di masa depan, sistem tidak memerlukan perantara manusia dan tersambung secara langsung ke sensor dan internet untuk mencatat data yang diambil dari dunia nyata.

2.2.3 UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak [6]. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Terdapat contoh diagram-diagram sebagai berikut :

2.2.3.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

2.2.3.2 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence* diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu

penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama [7].

2.2.3.3 *Sequence* Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktif hapusitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

2.2.3.4 *Activity* Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

2.2.3.5 *Class* Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja.

2.2.3.6 *Statechart* Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart* diagram).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari event tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

2.2.4 RFID

RFID (*Radio Frequency Identification*) merupakan teknologi digital dalam bentuk tag dan reader yang berbasis jaringan *wireless* (gelombang radio) untuk proses transfer data, identifikasi objek dan informasi elektrinik lainnya [7]. Gambar RFID *reader* dan *Tag* RFID dapat dilihat pada Gambar 2.9 RFID MFRC522.

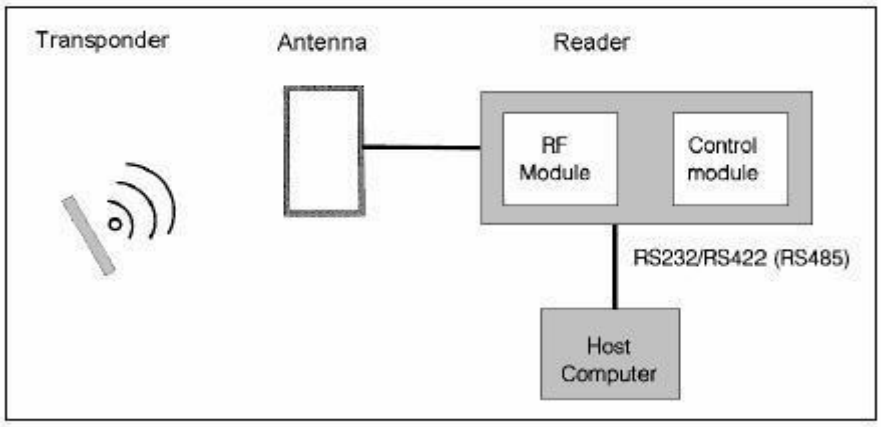
Gambar 2.2 RFID MFRC522

Teknologi RFID mudah digunakan, dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. RFID dapat disediakan dalam perangkat yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read*/*Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat

beroperasi, dapat berfungsi pada berbagai variasi lingkungan dan menyediakan tingkat integritasi yang tinggi.

Pada sistem RFID umumnya, *tag* atau *transponder* ditempelkan pada suatu objek. Setiap *tag* dapat membawa informasi yang unik, di antaranya: *serial number*, *model*, warna, tempat perakitan, dan data lain dari objek tersebut. Ketika *tag* ini melalui medan yang dihasilkan oleh pembaca RFID yang kompatibel, *tag* akan mentransmisikan informasi yang ada pada *tag* kepada pembaca RFID, sehingga proses identifikasi objek dapat dilakukan.

Sistem RFID terdiri dari empat komponen, di antaranya seperti dapat dilihat pada Gambar 2.3 Sistem RFID.



Gambar 2.3 Sistem RFID

1. *Tag* : Ini adalah *device* yang menyimpan informasi untuk identifikasi objek. *Tag* RFID sering juga disebut sebagai *transponder*.
2. Antena : untuk mentransmisikan sinyal frekuensi radio antara pembaca RFID dengan *tag* RFID.
3. Pembaca RFID (*Micro-Reader*): adalah alat yang kompatibel dengan *tag* RFID yang akan berkomunikasi secara *wireless* dengan *tag*.
4. *Software* aplikasi : adalah aplikasi pada sebuah *workstation* atau PC yang dapat membaca data dari tag melalui pembaca RFID. Baik *tag* dan pembaca RFID diperlengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

Walaupun teknologi RFID telah hadir selama hampir 20 tahun, belum ada standar data tunggal untuk satuan maupun untuk aplikasi industri. Sebagai tambahan terhadap biaya per label, ketiadaan suatu standar data yang jelas juga menjadi suatu faktor yang membatasi penggunaan RFID secara luas. Berdasarkan catu daya tag, tag RFID (*Radio Frequency Identification*) dapat digolongkan menjadi :

1. *Tag* Pasif : yaitu *tag* yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca RFID. Rangkaiannya lebih sederhana, harganya lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah tag hanya dapat mengirimkan informasi dalam jarak yang terbatas 4 - 5m ketika menggunakan frekuensi UHF (860 MHz– 930 MHz).
2. *Tag* Semi-Pasif : yaitu tag yang memiliki baterai terintegrasi dan oleh karena itu tidak memerlukan energi dari medan pembaca untuk menggerakkan *chip* itu. Ini memungkinkan *tag* untuk berfungsi dengan tingkatan sinyal yang lebih rendah, menghasilkan jarak yang lebih besar sampai kepada 100 meter. Jaraknya terbatas karena *tag* tidak mempunyai pemancar terintegrasi, dan masih perlu menggunakan medan pembaca untuk komunikasi kembali ke pembaca itu.
3. *Tag* Aktif : yaitu tag yang catu dayanya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh pembaca RFID dan *tag* dapat mengirimkan informasi dalam jarak yang lebih jauh (sampai beberapa kilometer). Kelemahan dari tipe *tag* ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih komplek. Semakin banyak fungsi yang dapat dilakukan oleh tag RFID maka rangkaiannya akan semakin komplek dan ukurannya akan semakin besar.

2.2.5 *Tag* RFID

Tag RFID telah sering dipertimbangkan untuk digunakan sebagai *barcode* pada masa yang akan datang. Pembacaan informasi pada *tag* RFID tidak memerlukan kontak sama sekali. Karena kemampuan rangkaian terintegrasi yang modern, maka tag RFID dapat menyimpan jauh lebih banyak informasi dibandingkan dengan *barcode*. Fitur pembacaan jamak pada teknologi RFID sering disebut sebagai *anti-collision*.

Pembagian kelas RFID adalah berdasarkan kemampuan untuk membaca dan menulis data. EPC (*Electronic Product Code*) global mengelompokkannya menjadi lima kelas yaitu :

1. Class 0 – *Read Only – Factory Programmed*

   Tipe ini adalah tipe yang paling sederhana, yang hanya mengandung nomor seri, EPC ditulis sekali ke dalam tag selama produksi. Datanya kemudian tidak dapat diubah lagi. Kelas 0 juga digunakan untuk menentukan kategori tag yag disebut EAS (*Electronic Artilce Surveilance*) atau alat anti pencuri, yang tidak mempunyai nomor seri, hanya mendeteksi keberadaannya saat melewati antena.
2. Class 1

   Dalam kelas ini, *tag* dihasilkan dengan tidak ada data yang ditulis ke dalam memori. Data kemudian bisa ditulis baik oleh pabrik *tag* atau oleh pemakai tetapi hanya sekali.

Selanjutnya tag tidak dapat ditulis lagi dan *tag* hanya dapat dibaca. *Tag* jenis ini umumnya digunakan sebagai identifikasi sederhana.

3. Class 2

   Tipe ini adalah tipe yang paling fleksibel, di mana pemakai mempunyai akses untuk membaca dan menulis data ke dalam memori tag. Tag ini biasa digunakan untuk mencatat data, dan oleh karena itu berisi lebih banyak memori dibanding dengan yang hanya digunakan untuk pengenal sederhana.

4. Class 3

   Tag ini berisi sensor terintegrasi untuk parameter perekaman seperti temperatur, tekanan, dan gerakan, yang dapat direkam dengan menulis ke dalam memori *tag*. Pembacaan sensor dapat diambil tanpa ada pembaca, *tag* dapat berupa *tag* aktif maupun *tag* semi-pasif.

5. Class 4

   Tipe ini seperti miniatur radio yang dapat berkomunikasi dengan *tag* dan alat lain tanpa adanya suatu pembaca. Ini berarti mereka dengan sepenuhnya aktif dengan sumber baterai mereka sendiri.

2.2.6 PHP

Personal Home Page atau sekarang biasa disebut PHP (*Hypertext Processor)* adalah *script* yang dijalankan di sisi *server* dengan konsep yang berbeda dengan *JavaScript* yang dijalankan pada sisi klien. Dalam penggunaan PHP, kode yang menyusun program tidak perlu dibagikan ke pemakai, yang berarti bahwa kerahasiaan kode dapat dilindungi sehingga menguntungkan bagi penggunanya [8].

Secara khusus, PHP dirancang untuk membentuk web dinamis yang dapat membentuk suatu tampilan berdasarkan permintaan terkini. Sebagai contoh, pengguna bisa menampilkan isi *database* ke halaman web. Pada prinsipnya, PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion* ataupun *Perl*.

Kenyataan bahwa PHP bisa digunakan untuk mengakses berbagai macam database seperti *Access, Oracle,* kode PHP dapat disisipkan pada kode HTML merupakan hal menarik yang didukung oleh PHP tetapi tidak mungkin dilakukan oleh *JavaScript*. Selain itu PHP juga bisa digunakan untuk menghasilkan kode-kode HTML. Logo PHP dapat dilihat pada Gambar 2.4 Logo PHP.

Gambar 2.4 Logo PHP

Kelebihan dari PHP, yaitu :

1. Bahasa pemoraman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunanya.
2. *Web server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lightpd, nginx, hingga Xitami dengan konfigurasi lebih mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan dibeberapa mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.2.7 MYSQL

MySQL adalah sebuah aplikasi *Relational Database Management Server* (RDBMS) bersifat *open source* yang memungkinkan data diakses dengan cepat oleh banyak pemakai secara bersamaan dan juga memungkinkan pembatasan akses pemakai berdasarkan *privilege* (hak akses) yang diberikan. MySQL menggunakan bahasa SQL (*structured query language*) yang merupakan bahasa standar pemograman database [9]. Logo *database* MySQL dapat dilihat pada Gambar 2.5 Logo MySQL.

Gambar 2.5 Logo MySQL

Berikut berbagai keunggulan yang ditemukan di MySQL :

1. Skalabilitas, MySQL dapat menangani database yang besar, yang implementasinya telah dibuktikan dalam organisasi seperti HP, Cisco,Yahoo, NASA, Google, dan lain sebagainya.
2. Portabilitas, MySQL dapat berjalan pada berbagai macam sistem operasi termasuk Unix, Solaris, Mac OS, Windows, Linux serta pada arsitektur yang berbeda, mulai dari *low-end* PC sampai *high-end mainframe*.
3. Konektivitas, MySQL sepenuhnya mendukung jaringan yang dapat diakses dari mana saja di internet serta pengguna dapat mengakses database MySQL secara bersamaan. MySQL juga menyediakan berbagai macam API (*Aplication Program Interface*) untuk mendukung konektivitas aplikasi yang ditulis dalam bahasa C, C++, C#, Perl, PHP, Java, Phyton, dan lain sebagainya.
4. Keamanan, MySQL mencakup seluruh keamanan yang kuat untuk mengontrol akses data dan juga mendukung *Secure Socket Layer* (SSL) *Protocol*.
5. Kecepatan, MySQL dikembangkan dengan kecepatan.
6. Mudah digunakan, MySQL mudah untuk digunakan dan diimplementasikan
7. *Open Source*, MySQLAB membuat kode MySQL tersedia untuk digunakan setiap orang. filosofi *open source* memungkinkan khalayak global untuk berpartisipasi dalam pengembangan.

Sebuah DBMS/RDBMS tidak dapat lepas dari SQL (*Structured Query Language*). SQL merupakan sebuah bahasa yang digunakan untuk mengelola dan berinteraksi dengan data dalam *database* relasional. SQL adalah bahasa *database* yang paling universal digunakan, dan itu telah menjadi bahasa standar untuk manajemen *database*. SQL bekerja sama dengan sebuah RDBMS untuk mendefinisikan struktur dari *database*, menyimpan data di *database* tersebut, memanipulasi data, mengambil data, mengontrol akses ke data, dan menjamin integritas data.

Berikut beberapa contoh perintah dasar SQL yang sering digunakan pada MySQL :

1. *Create* Database, perintah yang digunakan membuat database baru.
   Sintaks : CREATE DATABASE DATABASE_NAME
2. *Drop* Database, perintah yang digunakan untuk menghapus database.
   Sintaks : DROP TABEL TABEL_NAME
3. *Create* Tabel, perintah yang digunakan untuk membuat table baru.
   Sintaks : Create Tabel tabel_name (*create_definition*)
4. *Describe*, perintah yang digunakan untuk mendeskripsikan tabel.
   Sintaks : Describe (Desc) tabel [*column*]
5. *Alter* Tabel, perintah yang digunakan untuk menghapus tabel.

Sintaks : Alter [*Ignor*] Tabel table_name

6. *Drop* Tabel, perintah yang digunakan untuk menghapus tabel.
   Sintaks : Drop Tabel tabel_name [*tabel_name*..]
7. *Delete*, perintah yang digunakan untuk menghapus *record* dari tabel.
   Sintaks : Delete From tabel_name Where Where_definition
8. *Select*, perintah yang digunakan untuk query ke database.
   Sintaks : select * from tabel_name

2.2.8 XAMPP

XAMPP merupakan salah satu paket instalasi Apache, PHP, dan MySQL secara instan yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut yang memiliki fungsi sebagai *server* yang berdiri sendiri (localhost). Beberapa Fitur yang ada pada XAMPP antara lain FTP, MySQL, Apache, PHP, CGI-BIN, FTP, Mercury Mail (SMTP), Perl MyAdmin dan My Admin. Logo XAMPP dapat dilihat pada Gambar 2.6 Logo XAMPP .

Gambar 2.6 Logo XAMPP

Bagian penting dari XAMPP yang sering digunakan:

1. htdoct adalah folder tempat meletakkan file yang akan dijalankan, seperti file PHP, HTML dan script lainnya.
2. phpMyAdmin merupakan bagian untuk mengelola database MySQL pada browser. Dengan mengakses alamat http://localhost/phpMyAdmin, maka akan muncul halaman phpMyAdmin.
3. Kontrol Panel yang berfungsi untuk mengelola layanan (*service)* XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

2.2.9 *Web Browser*

Web Browser adalah *software* yang digunakan untuk menampilkan informasi dari *web server* yang telah dikembangkan dengan menggunakan *Graphic User Interface* (GUI), sehingga pengguna dapat dengan mudah melakukan *point* dan *click* untuk pindah antar dokumen. Web browser yang masih menggunakan mode teks adalah Lynx, yang

mengakibatkan tidak ada gambar yang dapat ditampilkan [10]. Biasanya Lynx terdapat pada lingkungan DOS dan Unix. Akan tetapi perkembangan dari *web browser* dengan GUI. Saat ini ada banyak *browser web* GUI, antara lain Mozilla Firefox, Internet Explorer, Opera, dan Google Chrome. Dengan berusahan mendekati standar dokumen HTML yang direkomendasikan oleh W3C, beberapa browser tersebut bersaing untuk merebut pemakainya.