BAB 2
LANDASAN TEORI

2.1. Sejarah Secure Parking

Secure parking didirikan sejak tahun 1979 di Sydney Australia oleh 2 (dua) bersaudara Garth Mathews & Brett Mathews. Secure Parking kin telah berkembang dan menjadi perusahaan penyedia jasa pelayanan pengelolaan perparkiran terbesar di Australia dan Indonesia. Secure Parking menjadi seperti saat ini Karena keinginan untuk selalu menjaga dan memperkuat hubungan kemitraan melalui setiap aspek pelayanan operasional kami.

Di Indonesia, Secure Parking telah melayani negeri ini sejak tahun 1992 dan telah memiliki lokasi parkir dalam operasional sebanyak 600 lokasi yang tersebar di beberapa kota besar di Indonesia, yakni Jabodetabek, Bandung, Yogyakarta, Surabaya, Malang, Medan, Batam, Pekanbaru, Palembang, Manado dan Jambi dengan total pengelolaan lebih dari 810.000 petak parkir dan didukung oleh lebih dari 12.000 putra putri Indonesia terpilih & terlatih.

2.1.1. Logo Secure Parking

Gambar 2.1 adalah logo dari Secure Parking yang sering digunakan



Gambar 2.1 Logo Secure Parking

2.1.2. Visi dan Misi Secure Parking

Adapun Visi dan Misi dari Secure Parking adalah sebagai berikut :

2.1.2.1. Visi Secure Parking

Menjadi perusahaan parkir termaju dan terkemuka dan mempunyai reputasi baik di Asia melalui sumber daya manusia yang mempunyai kompetensi tinggi

2.1.2.2. Misi Secure Parking

Menjalankan bisnis parkir berdasarkan inovasi dan menyelenggarakan prinsip - prinsip operasional yang terbaik, sumber daya manusia yang kompeten, memiliki hubungan keluar yang baik dengan semua pihak yang terkait.

2.2. Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori – teori yang berkaitan dalam pembangunan system monitoring dan control alat elektronik menggunakan *internet of things*. Beberapa teori terkait dengan pembangungn perangkat ini adalah konsep yang digunakan perangkat lunak, Perangkat Kearas dan Bahasa pemrograman yang dibutuhkan dalam pembangunan sistem pencarian tempat parkir ini.

2.2.1. Jaringan Komputer

Menurut definisi, jaringan komputer merupakan sejumlah komputer yang dapat saling berkomunikasi. Dalam komunikasi ini dapat terjadi perpindahan data ataupun berbagi sumber daya. Dalam skala luas, internet juga merupakan jaringan komputer. Jadi, suatu jaringan komputer tidak hanya terjadi pada sejumlah komputer yang terdapat pada suatu ruangan ataupun gedung atau perusahaan. Pada dasarnya tekonologi jaringan komputer itu sendiri merupakan perpaduan antara teknologi dan juga teknologi komunikasi **[1]**.

2.2.2. Internet of Things

Internet of Things (IoT) merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data remote control, dan termasuk juga pada benda dunia nyata. Bahan pangan, elektronik, peralatan apa saja, koleksi termasuk benda hidup, yang semuanya tersambung ke jaringan lokal dan global melalui sensor tertanam dan selalu “on” **[2]**.

Interaksi antara manusia dengan manusia sudah sangat biasa sejak zaman dahulu kala. Interaksi antara manusia dengan mesin sudah biasa pula, sejak adanya penemuan teknologi seperti komputer atau gadget *devices* lainnya. Namun, Menurut analisa McKinsey Global Institute, *internet of things* adalah sebuah teknologi yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen. Sebuah publikasi mengenai *Internet of things in 2020* menjelaskn bahwa *internet of things* adalah suatu keadaan ketika benda memiliki identitas, bisa beroperasi secara intelijen, dan bisa berkomunikasi dengan sosial, lingkungan, dan pengguna.

Dapat kita simpulkan bahwa *internet of things* membuat suatu koneksi antara mesin dengan mesin, sehingga mesin-mesin tersebut dapat berinteraksi dan bekerja secara independen sesuai dengan data yang diperoleh dan diolahnya secara mandiri. Tujuannya adalah untuk membuat manusia berinteraksi dengan benda dengan lebih mudah, bahkan supaya benda juga bisa berkomunikasi dengan benda lainnya.

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.

Cara kerja dari *internet of things* cukup mudah. Setiap benda harus memiliki sebuah IP Address. IP Address adalah sebuah identitas dalam jaringan yang membuat benda tersebut bisa diperintahkan dari benda lain dalam jaringan yang sama. Selanjutnya, IP address dalam benda-benda tersebut akan dikoneksikan ke jaringan internet. Saat ini, koneksi internet sudah sangat mudah kita dapatkan. Dengan demikian, kita dapat memantau benda tersebut bahkan memberi perintah kepada

benda tersebut. Sebagai contoh, jika ada speaker yang memiliki IP address dan terkoneksi internet di Amerika Serikat, kita dapat memerintahkan speaker tersebut untuk menyalakan musik walaupun kita berada di Indonesia. Yang kita perlukan hanyalah koneksi internet. Lalu apa hubungannya dengan *internet of things*, Setelah sebuah benda memiliki IP address dan terkoneksi dengan internet, pada benda tersebut juga dipasang sebuah sensor. Sensor pada benda memungkinkan benda tersebut memperoleh informasi yang dibutuhkan. Setelah memperoleh informasi, benda tersebut dapat mengolah informasi itu sendiri, bahkan berkomunikasi dengan benda-benda lain yang memiliki IP address dan terkoneksi dengan internet juga. Akan terjadi pertukaran informasi dalam komunikasi antara benda-benda tersebut. Setelah pengolahan informasi selesai, benda tersebut dapat bekerja dengan sendirinya, atau bahkan memerintahkan benda lain juga untuk ikut bekerja. Jadi, dalam *internet of things* manusia akan bertindak sebagai raja dan akan dilayani oleh benda-benda disekitarnya **[3]**.

2.2.3. Mikrokontroler

Mikrokontroler (pengendali mikro) pada suatu rangkaian elektronik berfungsi sebagai pengendali yang mengatur jalannya proses kerja dari rangkaian elektronik. Di dalam sebuah IC mikrontroler terdapat CPU, memori, timer, saluran komunikasi serial dan paralel, port input/output, ADC, dll. Mikrokontroler digunakan dalam sistem elektronik modern, seperti : Sistem manajemen mesin mobil, keyboard computer, Instrumen pengukuran elektronik (seperti multimeter digital, *synthesizer* fruekensi, dan osiloskop), televisi, radio, telepon digital, *mobile phone*, *microwave oven*, IP *Phone*, printer, scanner, kulkas, pendingin ruangan, CD/DVD player, kamera, mesin cuci, PLC (*Programmable Logic Controller*), Robot, sistem otomasi, sistem akuisisi data, sistem keamanan, peralatan medis (MRI, CT SCAN, ECG, EEG, USG), sistem EDC (*Electronic Data Capture*), mesin ATM, modem, router, dll **[4]**.

2.2.3.1. Sejarah Mikrokontroler

Sejarah mikrokontroler tidak terlepas dari sejarah mikroprosesor. Berikut ini sejarah mikroprosesor dan mikrokontroler :

1. Tahun 1971, Intel 4004 adalah Mikroprosesor pertama, Intel 4004 dibuat dan dikembangkan oleh Intel (*Integrated Electronics*). Intel membuat mikroprosesor Intel 4004 menggunakan 2250 transistor. Intel 4004 merupakan mikroprosesor 4 bit. Kemudian pada tahun 1974, Intel membuat mikroprosesor generasi kedua (Intel 8008), Intel 8008 merupakan mikroprosesor 8 bit.



Gambar 2.2 Intel 4004

2. Tahun 1972, TMS1000 adalah mikrokontroler pertama. TM1000 merupakan mikrokontroler 4-bit. Mikrokontroler TM1000 dibuat oleh Gary Boone dari Texas Instrument.



Gambar 2.3 TMS 1000

3. Tahun 1974, beberapa pabrikan IC menawarkan mikroprosesor dan pengendali menggunakan mikroprosesor. Mikroprosesor yang

ditawarkan pada saat itu yaitu Intel 8080, 8085, Motorola 6800, Signetics 6502, Zilog Z80, dan Texas Instruments 9900 (16 bit).

4. Tahun 1976, dibuat Intel 8048, yang merupakan mikrokontroler intel pertama
5. Tahun 1978, mikroprosesor 16 bit menjadi umum digunakan yaitu Intel 8086, Motorola 68000 dan Zilog Z8000. Sejak saat itu pabrikan mikroprosesor terus mengembangkan mikroprosesor dengan berbagai keistimewaan dan arsitektur. Mikroprosesor yang dikembangkan termasuk 32 bit device seperti Intel Pentium, Motorola DragonBall, dan beberapa mikrokontroler yang menggunakan ARM *core*.
6. Tahun 1980, Intel 8051 atau lebih dikenal dengan mikrokontroler MCS51. MCS51 adalah mikrokonroler CISC 8 bit.
7. Tahun 1996, mikrokontroler Atmel AVR dibuat oleh Atmel. Atmel AVR adalah salah satu mikrokontroler yang banyak digunakan sampai dengan saat ini. AVR adalah mikrokontroler RISC (*Reduce Instruction Set Computing*) 8 bit berdasarkan arsitektur Harvard.

2.2.3.1. Cara Kerja Mikrokontroler

Prinsip kerja mikrokontroler adalah sebagai berikut:

1. Berdasarkan nilai yang berada pada register Program Counter, mikrokontroler mengambil data dari ROM dengan address sebagaimana nilai yang tertera pada Program Counter. Selanjutnya Program Counter ditambah nilainya dengan 1 (increment) secara otomatis. Data yang diambil tersebut adalah urutan intruksi program pengendali mikrokontroler yang sebelumnya telah dibuat oleh pemakai.
2. Intruksi tersebut diolah dan dijelaskan. Proses pengerjaan bergantung pada jenis intruksi: bisa membaca, menguba nilai – nilai pada register, RAM, isi port, atau melakukan pembacaan dan dilanjutkan dengan pengubahan data.

3. Program Counter telah berubah nilainya (baik karena penambahan otomatis sebagimana pada langkah 1 diatas atau karena pengubahan pada langkah 2). Selanjutnya yang dilakukan mikrokontroler adalah mengulang kembali siklus ini pada langkah 1 dan seterusnya.

Dari pengertian diatas dapat disimpulkan bahwa pada dasarnya untuk kerja mikrokontroler sangatlah bergantung pada urutan intruksi yang dijalankanya, yaitu program yang ditulis di ROM **[5]**.

2.2.4. Arduino

Arduino adalah perangkat elektronik yang berfungsi seperti mikrokontroler dan bersifat *open source*. Arduino memiliki prosesor keluarga Atmel AVR. Arduino memiliki perangkat lunak dengan bahasa pemograman yang spesifik. Arduino juga memiliki *sofware* kompilasi sendiri yang bersifat *open source* dan dapat diunduh di website *arduino.cc*. Sehingga pengguna dapat mengembangkan sendiri *board* Arduino sesuai dengan keinginan.

Board Arduino mempunyai kemampuan untuk membaca masukan data digital dan data analog. Board Arduino juga mempunyai kemampuan untuk mengeluarkan data digital dari analog. Bahasa pemrogaman yang digunakan mempunyai sintaks bahasa C. Pengguna Arduino dapat mengembangkan beberapa proyek sederhana (motor, led data serial) dan proyek yang kompleks (*Internet of Things*, printer 3D dan aplikasi canggih lainya). Arduino dapat berjalan diatas komputer dengan sistem oprasi mac, windows dan linux **[6]**.

2.2.4.1. Sejarah Arduino

Arduino dikembangkan dari thesis Hernando Barragan pada tahun 2004, seorang mahasiswa asal kolombia. Judul thesisnya yaitu "Arduino-Revolusi *Open Hardware*". Arduino diawali dari ruangan kelas *Interactive Design Institute* di Ivrea (IDII), pada tahun 2005 di Ivera, italia. Arduino ditemukan oleh Massimo Banzi dan David Cuartielles dengan tujuan awal yaitu untuk membantu para siswa membuat perangkat desain dan interaksi dengan harga yang murah dibandingkan dengan perangkat lain yang tersedia pada saat itu, seperti *BASIC Stamp* yang harganya cukup mahal bagi pelajar pada saat itu. Arduino berasal dari bahasa Italia yang berarti teman yang berani. Pada bulan Mei 2011,arduino sudah terjual

lebih dari 3000 unit arduino saat ini sudah menjadi salah satu *platform* OSHW (*Open Source Hardware*). Arduino memiliki banyak jenis, berikut ini adalah jenis – jenis arduino yaitu:

1. Arduino Uno (R3)
2. Arduino Due
3. Arduino Mega2560
4. Arduino Leonardo
5. Arduino Fio
6. Arduino LilyPad
7. Arduino Nano
8. Arduino Mini
9. Arduino Micro
10. Arduino Ethernet
11. Arduino Espora
12. Arduino Robot

2.2.5. Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan peralngkan lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientas objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek (A.S. & Shalahuddin, 2011).

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekata terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut. Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (reusable).
2. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.
3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunai nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.2.5.1. Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dihubungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancanggan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu (A.S. & Shalahuddin, 2011).

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (Class)

   Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (relationship) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara tekniks kelas adalah sebuah struktur dalam pembuatan perangkat

lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (object)

Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satua organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstra. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan. Secara teknis, sebuah kelas saat program dieksekusi makan akan dibuat sebuah objek. Objek dilihat darisegi teknis adalah elemen pada saat runtime yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehinga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (method)

Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari event, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunnia nyata. Contoh metode atau operasi misalnya Read, Write, Move, Copy, dan sebagainya.

4. Atribut (attribute)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat beruoa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (abstraction)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (encapsulation)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (inheritance)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (interface)

Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain.

9. Reusability

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

Komunikasi antarobjek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (polymorphism)

Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. Package

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda **[7]**.

2.2.5.2. Pengenalan UML

UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk menvisualisasi, menspesifikasi, merancang dan mendokumentasi sistem piranti lunak. Unified Modelling Language (UML) menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan UML adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan. UML menyediakan beberapa notasi dan artifact standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Arti fact didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman **[8]**.

2.2.5.3. Diagram UML (*Unified Modeling Language*)

UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan singkatan dari Unified Modeling Language. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu developer ke developer lainya. Tidak hanya antar developer terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML **[7]**.

2.2.5.4. *Use Case* Diagram

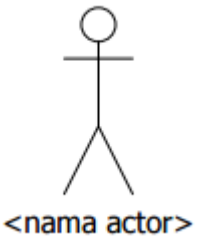
Use Case atau diagram Use Case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, Use Case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi - fungsi itu.

Syarat penamaan pada Use Case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada Use Case yaitu pendefinisian apa yang disebut aktor dan Use Case.

1. *Actor*

Actor adalah sesuatu (entitas) yang berhubungan dengan sistem dan berpartisipasi dalam use case. *Actor* menggambarkan orang, sistem atau entitas Eksternal yang secara khusus membangkitkan sistem dengan input atau masukan kejadian-kejadian, atau menerima sesuatu dari sistem. *Actor* dilukiskan dengan peran yang mereka mainkan dalam *use case*, seperti Staff, Kurir dan lain-lain.

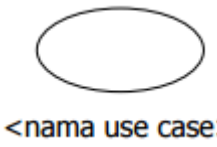
Dalam *use case* diagram terdapat satu *actor* pemulai atau initiator *actor* yang membangkitkan rangsangan awal terhadap sistem, dan mungkin sejumlah aktor lain yang berpartisipasi atau *participating actor*. Akan sangat berguna untuk mengetahui siapa *actor* pemulai tersebut **[8].**



Gambar 2.4 Bentuk *Actor* dalam UML

2. *Use Case*

Use case yang dibuat berdasarkan keperluan aktor merupakan gambaran dari "apa" yang dikerjakan oleh sistem, bukan "bagaimana" sistem mengerjakannya. *Use case* diberi nama yang menyatakan apa hal yang dicapai dari interaksinya dengan aktor.

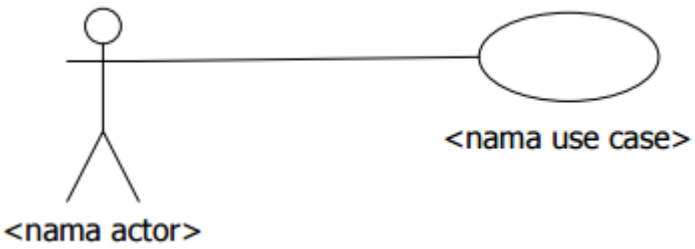


Gambar 2.5 Bentuk *Use Case*

3. *Relationship*

Relasi (relationship) digambarkan sebagai bentuk garis antara dua simbol dalam use case diagram. Relasi antara actor dan use case disebut juga dengan asosiasi (association). Asosiasi ini digunakan untuk menggambarkan bagaimana hubungan antara keduanya.

Relasi-relasi yang terjadi pada use case diagram bisa antara actor dengan use case atau use case dengan use case.



Gambar 2.6 Bentuk *Relationship* dalam UML

Relasi antara use case dengan use case :

1. Include, pemanggilan use case oleh use case lain atau untuk menggambarkan suatu use case termasuk di dalam use case lain (diharuskan). Contohnya adalah pemanggilan sebuah fungsi program. Digambarkan dengan garis lurus berpanah dengan tulisan <<include>>.

2. Extend, digunakan ketika hendak menggambarkan variasi pada kondisi perilaku normal dan menggunakan lebih banyak kontrol form dan mendeklarasikan ekstension pada use case utama. Atau dengan kata lain adalah perluasan dari use case lain jika syarat atau kondisi terpenuhi. Digambarkan dengan garis berpanah dengan tulisan <<extend>>.
3. Generalization/Inheritance, dibuat ketika ada sebuah kejadian yang lain sendiri atau perlakuan khusus dan merupakan pola berhubungan baseparent use case. Digambarkan dengan garis berpanah tertutup dari base use case ke parent use case.

2.2.5.5. Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

2.2.5.6. *Sequence* Diagram

Diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sequence yang harus digambar adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use

case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sequence sehingga semakin banyak use case yang didefinisikan maka diagram sequence yang harus dibuat juga semakin banyak.

Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.

2.2.5.7. *Class* Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

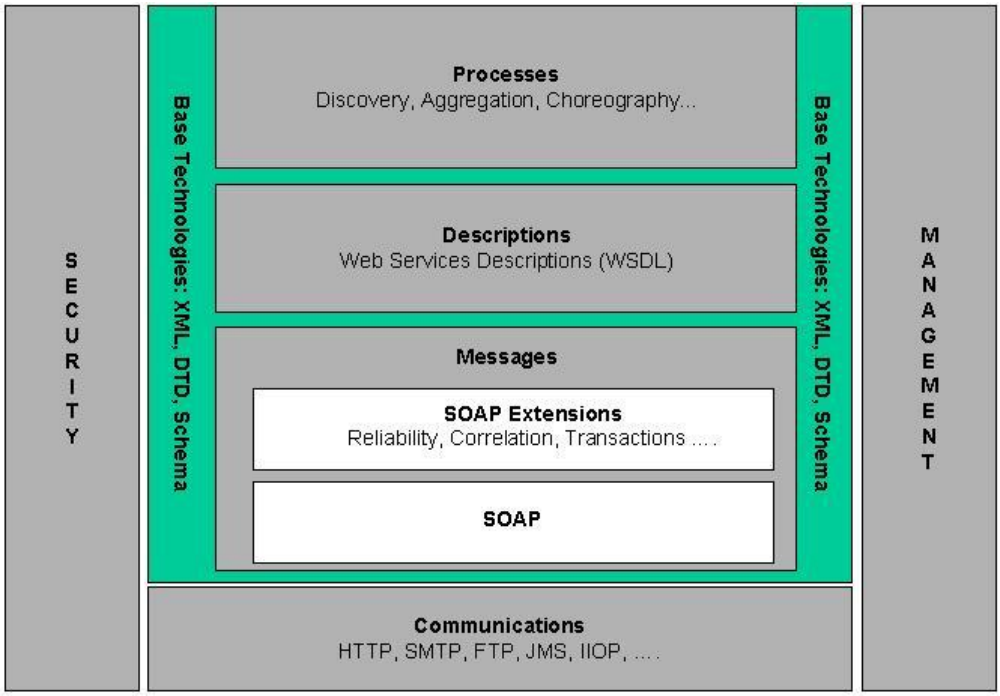
Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsifungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas Main Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kelas yang diambil dari pendefinisian *usecase* Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case.
4. Kelas yang diambil dari pendefinisian data Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca file teks, dan lain sebagainya sesuai kebutuhan **[9]**.

2.2.6. *Web Service*

Web service adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam platform dan antarmuka bahasa pemograman yang independen. Tujuan pengembangannya adalah untuk menjembatani komunikasi antar program, sehingga aplikasi yang satu dan aplikasi yang lain yang terdapat pada suatu jaringan yang sama atau pada jaringan berbeda dapat saling berkomunikasi asalkan menggunakan standart protokol yang ditetapkan oleh *web service*. Hal ini bisa terjadi, karena standart protokol itu tidaklah terikat pada suatu platform atau bahasa pemograman.

Protokol itu sendiri dibangun oleh *Extensible Markup Language* (XML) yang memang kenyataannya telah didukung oleh banyak paltform, bahasa pemograman, dan oleh developer di seluruh dunia. Arsitektur web service dibangun oleh beberapa layer dan teknologi yang saling berhubungan. Banyak cara untuk memvisualisaskan *service*, sama banyaknya dengan cara untuk membangun dan menggunakan *web service* gambar berikut hanyalah salah satu cara untuk menggambarkan arsitekturnya.



Gambar 2.7 Arsitektur *Web Service*

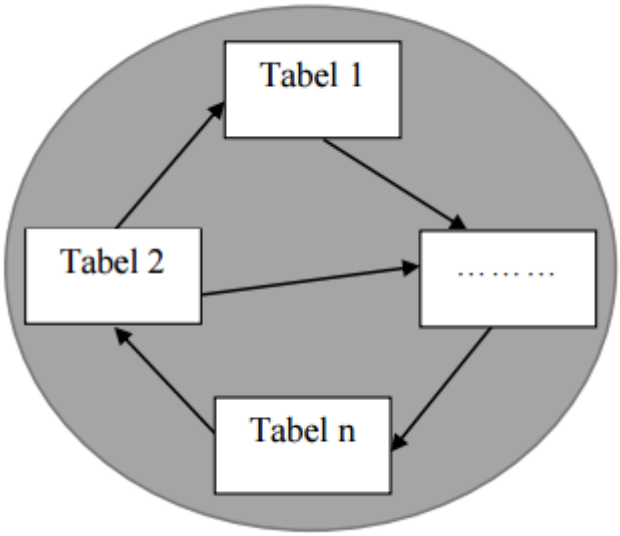
Web service disusun oleh tiga komponen standart, yaitu :

1. *Simple Object Access Protocol* (SOAP), yaitu protokol yang bertanggungjawab dalam pertukuran informasi dalam lingkungan jaringan terdistribusi.
2. *Web service definition Language* (WDSL), dokumen standart yang dituliskan pada format XML, dan mendefinikasan kehadiran web service dalam suatu jaringan.
3. *Universal description, Discovery and Inttegration*(UDDI), yaitu suatu lokasi direktori yang berisikan service (layanan) dan bersifat bebas platform (platform independent), dituliskan berbasis XML dan dapat diakses oleh entitas yang berada di dalam dan luar jaringan.

Adanya standart tersebut membuat web service mudah di akses melalui berbagai antarmuka dan juga memberi peluang dimungkinkannya berbagai sistem yang dibangun pada platform yang berbeda dan bahasa yang berbeda untuk berkolaborasi dalam suatu pekerjaan.

2.2.7. Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Pada buku ini menggunakan basis data relasional yang diimplementasikan dengan table-tabel yang saling memiliki relasi seperti pada gambar berikut



Gambar 2.8 Gambaran Basis Data

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa file teks ataupun Database Management System (DBMS). Kebutuhan basis data dalam sistem informasi meliputi:

1. Memasukan, menyimpan dan mengambil data.
2. Membuat laporan berdasarkan data yang telah disimpan.

Tujuan dari buatnya table-tabel disini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola piker penyimpanan data nantinya jika dalam bentuk baris-baris data (record) dimana setiap baris terdiri dari beberapa kolom.

2.2.7.1. Database MySQL

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user. MySQL adalah implementasi dari system manajemen basisdata

relasional (RDBMS). MySQL dibuah oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri

MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael "Monty". Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL (General Public License) namun tidak boleh menjadikan produk turunan yang bersifat komersial. Pada saat ini MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database yaitu SQL. SQL (Structured Query Language) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan database lebih user-friendly dibandingan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni. SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemograman seperti C, dan Delphi.

Elemen dari SQL yang paling dasar antara lain pernyataan, nama, tipe data, ekspresi, konstanta dan fungsi bawaan. Pernyataan Perintah dari SQL yang digunakan untuk meminta sebuah tindakan kepada DBMS.

Sistem database menyediakan bahasa untuk mendefinisikan database Data Definition Language (DDL) serta bahasa untuk memanipulasi database Data Manipulation Language (DML) untuk melakukan operasi-operasi tertentu pada database dan juga Data Control Language (DCL).

a. Data Definition Language (DDL)

   Kumpulan perintah SQL yang digunakan untuk membangun database, bisa juga memiliki arti lain perintah untuk membuat struktur dasar database dan tabel. Perintah – perintah yang termasuk DDL adalah statement connect, create, show, describe, drop, rename dan alter atau seperti contoh berikut :

1. CREATE DATABASE nama_database;
2. SHOW database;
3. USE nama_database;

b. Data Manipulation Language (DML)

Kumpulan statement yang digunakan sebagai perintah untuk mengelola data dalam sebuah tabel. Perintah DML digunakan setelah menggunakan DDL antara lain seperti statement insert, update, delete dan select atau seperti contoh berikut :

1. INSERT INTO nama_tabel VALUES isi_field;
2. SELECT * FROM nama_tabel;
3. UPDATE nama_tabel SET FIELD = nilai WHERE kondisi;

2.2.8. Algoritma A*

Algoritma A* (A-Star) merupakan salah satu dari algoritma yang menerapkan teknik heuristik (teknik pencarian terbimbing). Algoritma A* (A-Star) merupakan salah satu algoritma yang dipelajari untuk menyelesaikan permasalahan. Algoritma A* menyelesaikan masalah yang menggunakan graf untuk perluasan ruang statusnya. Dengan kata lain digunakan untuk menyelesaikan permasalah yang bisa direpresentasikan dengan graf. Algoritma A* adalah sebuah algoritma yang telah diperkaya dengan menerapkan suatu heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan.

A* adalah algoritma Best First Search yang merupakan perpaduan *Uniform Cost Search* yang memilih jarak paling kecil dari simpul awal ke simpul berikutnya dan *Greedy-Best First Search* yang menggunakan nilai heuristik atau nilai perkiraan untuk menentukan simpul berikutnya. Oleh karena itu rumus dari algoritma ini adalah

$f(\boldsymbol{n}) = \boldsymbol{g}(\boldsymbol{n}) + \boldsymbol{h}\,(\boldsymbol{n})$…….(1) dimana,

$f(n) : biaya\ yang\ dibutuhkan$

$g(n) : biaya\ yang\ ditempuh\ dari\ node\ asal$

$h(n) : nilai\ perkiraan\ dari\ node\ saat\ ini\ ke\ tujuan$

Algoritma A* ini akan menemukan rute yang complete (selalu menemukan solusi jika ada) dan optimal.

Beberapa terminologi dasar yang terdapat pada algoritma ini adalah *starting point*, simpul(*nodes*), A, *open list*, *closed list*, harga (*cost*), Rintangan (*unwalkable*) penjelasan dari beberapa terminologi dapat dilihat di bawah ini

Starting point adalah sebuah terminologi untuk posisi awal sebuah benda. A adalah simpul yang sedang di jalankan dalam algoritma pencarian jalan terpendek. Simpul adalah petak-petak kecil sebagai representasi dari area *pathfinding.* Bentuknya dapat berupa persegi, lingkaran, maupun segitiga. *Openlist* adalah tempat menyimpan data simpul yang mungkin diakses dari *starting point* maupun simpul yang sedang dijalankan.. *Closed list* adalah tempat menyimpan data simpul sebelum A yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan. Rintangan (*unwalkable*) adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh A.

Harga (F) adalah nilai yang di peroleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke A, dan H, jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan. Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal (*starting point*) menuju simpul tujuan dengan memperhatikan harga (F) terkecil[12].

2.2.8.1. Fungsi Heuristik *Manhattan Distance*

Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*). Fungsi Heuristik digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan .

A star sebagai algoritma pencarian yang menggunakan fungsi heuristic untuk pencarian rute *node-node* pada peta. Ada beberapa fungsi heuristik umum yang bisa dipakai untuk algoritma A star ini, salah satunya adalah *Manhattan distance*. Rumus dari heuristik *Manhattan distance* adalah sebagai berikut :

h(n) = D * (abs(n.x-goal.x) + abs(n.y-goal.y))

Keterangan :

D = Pergerakan nilai g

abs = *Absence* (Ketiadaan)

Dimana h(n) adalah sebuah fungsi heuristik. D (distance) adalah jarak dari *node* awal (starting point) menuju *node* tujuan (goal).

2.2.9. Perangkat Keras

Perangkat keras adalah suatu perangkat yang dapat membantu mempermudah dalam pembuatan alat. Perangkat keras berupa perangkat yang berbentuk fisk. Perangkat keras yang digunakan antara lain adalah :

2.2.9.1. Arduino Uno

Board Arduino Uno menggunakan mikrokontroler ATmega328. Secara umum posisi/letak pin-pin terminal I/O pada berbagai Board Arduino posisinya sama dengan posisi/letak pin-pin terminal I/O dari Arduino Uno yang mempunyai 14 pin Digital yang dapat di set sebagai *Input/Output* (beberapa diantaranya mempunyai fungsi ganda), 6 pin Input Analog **[6]**.



Gambar 2.9 Board Arduino Uno

Berikut adalah spesifikasi dari Arduino Uno :

Tabel 2.1 Spesifikasi Arduino Uno

Deskripsi	Detail
Chip Mikrokontroller	ATmega328
Tegangan Operasi	5v
Tegangan Input	7V-12V
Digital I/O pin	14 buah, 6 diantaranya PWM
Analog Input pin	6 buah
Arus DC per pin I/O	40 mA
Arus DC pin 3.3V	50 mA
Memori Flash	32 KB, 0,5 KB digunakan booatloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 Mhz
Dimensi	68,6 mm x 53,4 mm
Berat	25

2.2.9.2. Sensor Ultrasonik

Sensor Ultrasonik bekerja dengan cara memancarkan suatu gelombang dan kemudian meghitung waktu pantulan gelombang tersebut. Gelombang ultrasonic bekerja pada frekuensi mulai dari 20 KHz sampai dengan 20 MHz. Frekuensi kerja yang digunakan dalam gelombang ultrasonic bervariasi tergantung pada medium yang dilalui, mulai dari kerapatan pada fasa gas, cair, hingga padat.

Sensor ultrasonic terdiri dari sebuah chip pembangkit sinyal 40 KHz, sebuah speaker ultrasonic, dan sebuah microphone ultrasonic. Speaker ultrasonic megubah sinyal 40 KHz menjadi suara sementara microphone ultrasonic berfungsi untuk mendeteksi pantulan suaranya. Sensor ultrasonic akan mengirimkan suara ultrasonic ketika ada pulsa trigger dari mikrokontroler **[6]**.



Gambar 2.10 Sensor Ultrasonik

Sensor ultrasonic buatan Parallax (Sensor PING) dapat digunakan untuk mengukur jarak sejauh 2 cm sampai 300 cm. Karakteristik dari sensor ultrasonic "PING" adalah :

Tabel 2.2 Karakteristik Sensor Ultrasonik

Deskripsi	Detail
Tegangan Supply	5 Voc
Konsumsi Arus	30 mA (maksimum 35 mA)
Jarak	2 cm sampai dengan 4m
Input Trigger	Pulsa TTL positif, minimal 2 uS, 5 uS typical
Echo Pulse	Pulsa TTL, positif, 115 uS sampai dengan 18.5

Deskripsi	Detail
	ms
Echo Hold-Off	750 uS
Frekuensi Burst	40 KHz untuk 200 uS
Delay untuk pengukuran selanjutnya	Minimal 200 uS

2.2.9.3. LCD (*Liquid Crystal Display*) 16x2

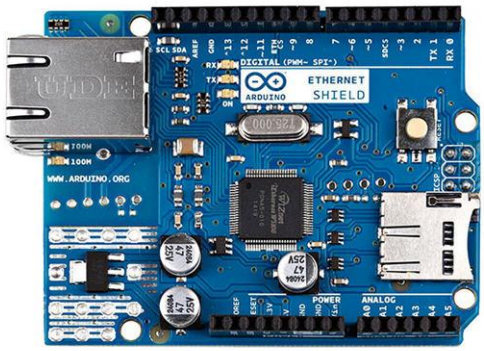
LCD (*liquid crystal display*) 16x2 adalah *display* dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dan matriks. LCD (*liquid crystal display*) 16x2 dapat menampilkan sebanyak 32 karakter yang terdiri dari 2 baris dan tiap baris dapat menampilkan 16 karakter **[6].**



Gambar 2.11 LCD 16x2

2.2.9.4. Ethernet Shield

Ethernet Shield digunakan untuk komunikasi data melalui jaringan berbasis TCP/IP. *Ethernet Shield* berfungsi untuk pengendalian dan monitoring melalui internet. Modul *Ethernet Shield* di hubungkan ke *board* Arduino melalui *port* SPI Arduino. Modul *Ethernet Shield* di hubungkan ke jaringan computer menggunakan kabel RJ45 **[6]**.



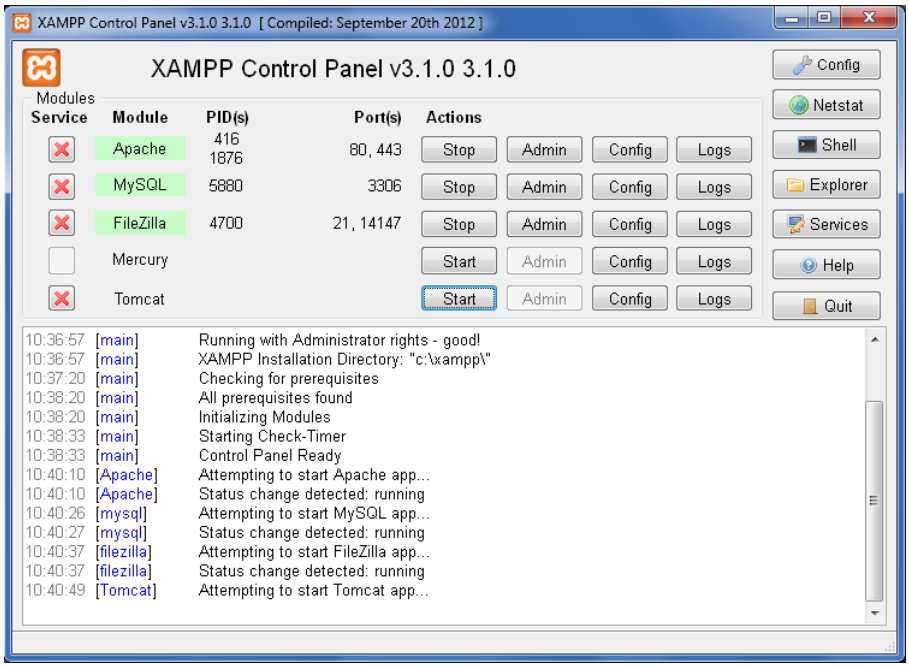
Gambar 2. 12 Ethernet Shield

2.2.10. Perangkat Lunak

Adapun beberapa perangkat lunak pendukung dalam penyususan penelitian pembangunan sistem pencarian tempat parkir cerdas otomatis menggunakan *internet of things* ini adalah

2.2.10.1. Xampp

Xampp yang berfungsi sebagi server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database dan penerjemah Bahasa yang ditulis dengan Bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Untuk mendapatkanya XAMPP anda dapat mendownload langsung dari web resminya. Dan berikut beberapa definisi program lainnya yang terdapat dalam XAMPP.



Gambar 2. 13 XAMPP Control Panel

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi seperti (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

phpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui WWW (World Wide Web). phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (fields), relasi (relations), indeks, pengguna (users), perijinan (permissions), dan lain-lain).Pada

dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (command line) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (database), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

2.2.10.2. Arduino IDE

Arduino IDE adalah software yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengeditprogram dalam bahasa Processing.
2. Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasaProcessing. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
3. Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino. Sebuah kode program Arduino umumnya disebut dengan istilah sketch. Kata “sketch” digunakan secara bergantian dengan “kode program” dimana keduanya memiliki arti yang sama.



Gambar 2. 14 Tampilan Pertama Arduino IDE

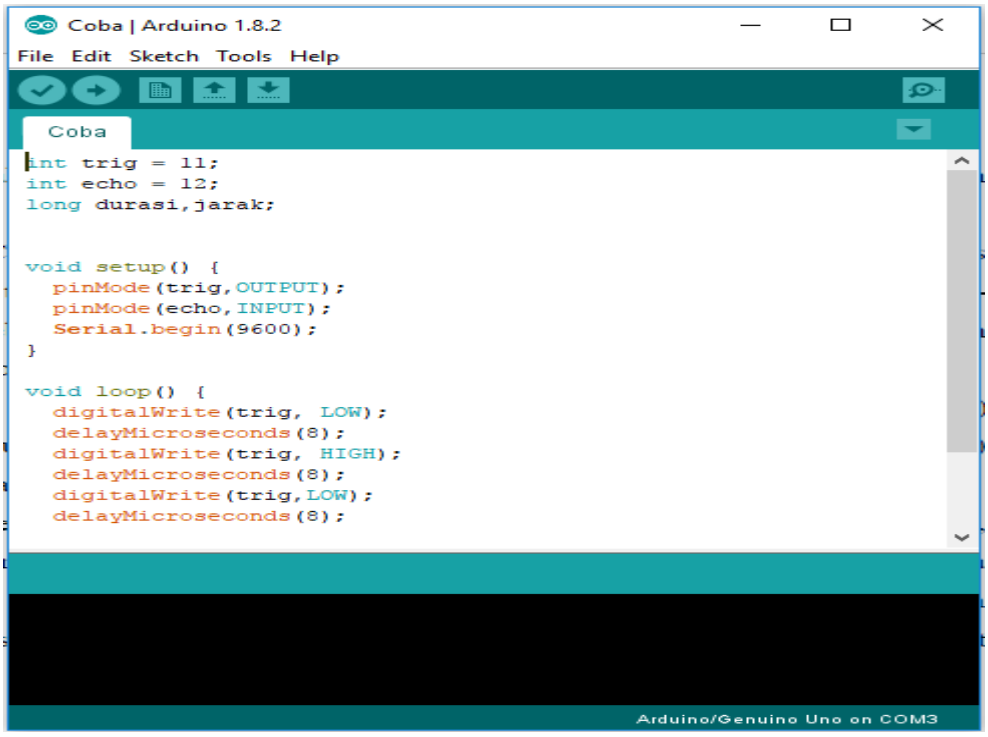
1. Bahasa Pemrograman Arduino Berbasis Bahasa C

Seperti yang telah dijelaskan diatas program Arduino sendiri menggunakan bahasa C. walaupun banyak sekali terdapat bahasa pemrograman tingkat tinggi (high level language) seperti pascal, basic, cobol, dan lainnya. Walaupun demikian, sebagian besar dari paraprogramer profesional masih tetap memilih bahasa C sebagai bahasa yang lebih unggul, berikut alasan-alasannya:

1. Bahasa C merupakan bahasa yang powerful dan fleksibel yang telah terbukti dapat menyelesaikan program-program besar seperti pembuatan sistem operasi, pengolah gambar (seperti pembuatan game) dan juga pembuatan kompilator bahasa pemrograman baru.
2. Bahasa C merupakan bahasa yang portabel sehingga dapat dijalankan di beberapa sistem operasi yang berbeda. Sebagai contoh program yang kita tulis dalam sistem operasi windows dapat kita kompilasi didalam sistem operasi linux dengan sedikit ataupun tanpa perubahan sama sekali.
3. Bahasa C merupakan bahasa yang sangat populer dan banyak digunakan oleh programer berpengalaman sehingga kemungkinan besar library

pemrograman telah banyak disediakan oelh pihak luar / lain dan dapat diperoleh dengan mudah.

4. Bahasa C merupakan bahasa yang bersifat modular, yaitu tersusun atas rutin-rutin tertentu yang dinamakan dengan fungsi (function) dan fungsi-fungsi tersebut dapat digunakan kembali untuk pembuatan program-program lainnya tanpa harus menulis ulang implementasinya.
5. Bahasa C merupakan bahasa tingkat menengah (middle level language) sehingga mudah untuk melakukan interface (pembuatan program antar muka) ke perangkat keras.
6. Struktur penulisan program dalam bahasa C harus memiliki fungsi utama, yang bernama main(). Fungsi inilah yang akan dipanggil pertama kali pada saat proses eksekusi program. Artinya apabila kita mempunyai fungsi lain selain fungsi utama, maka fungsi lain tersebut baru akan dipanggil pada saat digunakan.



Gambar 2. 15 Tampilan Arduino IDE

2.2.10.3. PHP

PHP adalah suatu bahasa server-side yang didesain khusus untuk aplikasi web. PHP dapat disisipkan diantara HTML, dan karena bahasa server-side, maka bahasa PHP akan dieksekusi di server, yang dikirimkan ke browser adalah "hasil jadi" bentuk HTML, dan kode PHP tidak akan terlihat. PHP dahulu merupakan proyek pribadi dari Rasmus Lerdorf tahun 1994 (dengan dikeluarkannya PHP versi 1 yang digunakan untuk membuat home page pribadi yang dipublikasikan 8 juni 1995). Versi pertama ini berupa kumpulan script PERL. Untuk versi kedua Ramus menulis ulang script-script PERL tersebut menggunakan bahasa C, kemudian menambahkan fasilitas untuk form HTML dan koneksi MYSQL. Adapun PHP didapat dari singkatan PHP Hypertext Preprocessor.

Setelah mengalami perkembangan, Zeev Suraski dan Andi Gutmans (termasuk Rasmus) yang tergabung dalam suatu kelompok open source, maka mereka memulai versi 3 PHP nya, PHP telah menampakkan keunggulan sebagai salah satu bahasa server scripting yang handal. Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan oleh kelompok ini PHP disebut sebgai "PHP: Hypertext Preprocessor". Pada bulan Mei 2000 dirilis PHP versi 4 yang dikeluarkan oleh Zend Engine yaitu sebuah produk yang diciptakan oleh Suraski dan Gutmans. Kehandalan PHP diperlengkap dengan dirilisnya PHP versi 5 pada 13 Juli 2004 yang mendukung OOP serta berbagai manajemen database yang luas dan penanganan error yang mudah. Sintak yang digunakan berasal dari bahsa C, Java maupun PERL. Sampai tulisan ini dibuat versi PHP yang terbaru adalah versi 5.2.0 yang dirilis pada November 2006.

PHP merupakan bahasa script yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan. Oleh karena itu, spesifikasi server lebih berpengaruh pada eksekusi dari script php daripada spesifikasi client. Namun tetap diperhatikan bahwa web yang dihasilkan tentunya harus dapat dibuka oleh

browser pada client. Dalam hal ini versi dari html yang digunakan harus didukung oleh browser client. PHP termasuk dalam Open Source Product. Jadi dapat menambah source code dan dapat mendistribusikannya secara bebas.