

BAB 2

LANDASAN TEORI

2.1 Profil Perusahaan

Profil SMKN 1 Majalaya meliputi sejarah, logo, visi, misi, tujuan, struktur organisasi dan deskripsi kerja SMKN 1 Majalaya.

2.1.1 Sejarah SMKN 1 Majalaya

Sekolah Menengah Kejuruan Negeri 1 Majalaya mulai berdiri pada bulan juni 2007, yang beroperasi pada tahun pelajaran 2007-2008, dengan menempati tempat belajar menumpang di SMP Negeri 1 Majalaya selama 1 tahun, membuka dua program keahlian dengan jumlah siswa 81 orang pada program keahlian Teknik Komputer Jaringan dan program keahlian Teknik Elektronika Industri dengan jumlah siswa 18 orang, sehingga jumlah siswa keseluruhan adalah 99 orang.

Pada tahun pelajaran 2008-2009 lokasi sekolah pindah ke Balekambang, tepatnya mengontrak di sebuah bangunan bekas sekolah swasta milik Yayasan Pendidikan Syukur 8 Majalaya yang berlokasi di Desa Sukamaju Kecamatan Majalaya, pada tahun kedua SMK Negeri 1 Majalaya memiliki jumlah siswa sebanyak 226 orang.

Pada tahun ke tiga berdirinya SMK Negeri 1 Majalaya, masih menempati lokasi di Balekambang, dipimpin oleh seorang Kepala Sekolah bernama Drs. Samsudin Hermawan, yang merupakan Kepala Sekolah Pelaksana Harian (Plh) karena SMK Negeri 1 Majalaya masih menginduk kepada SMK Negeri 2 Baleendah yang pada saat itu Kepala Sekolahnya dijabat oleh Drs. Asep Rusmana. Pada tahun itu SMK Negeri 1 Majalaya memiliki total jumlah murid sebanyak 452 orang yang terdiri dari 178 orang di Jurusan Teknik Elektronika Industri dan 274 orang di Jurusan Teknik Komputer Jaringan.

Pada tahun pelajaran 2009-2010, barulah sekolah ini memiliki bangunan sendiri dengan jumlah ruang kelas sebanyak 8 (delapan) ruang, 2 ruang praktek untuk bengkel Teknik Elektronika Industri dan bengkel praktek Teknik Komputer Jaringan, satu ruang Tata Usaha dan guru serta satu ruangan Kepala Sekolah, pada tahun pelajaran 2009-2010 lulusan pertama SMK Negeri 1 Majalaya dilepas sebanyak 96 orang yang sekarang bekerja di perusahaan-perusahaan daerah dan sebagian lagi melanjutkan Pendidikan ke Perguruan Tinggi.

2.1.2 Logo SMKN 1 Majalaya

Logo SMK Negeri 1 Majalaya dapat dilihat pada Gambar 2.1



Gambar 2.1 Logo SMK Negeri 1 Majalaya

Arti logo SMK Negeri 1 Majalaya:

Warna biru muda	: Perlambangan Kewibawaan
Padi dan Kapas	: Kesejahteraan dan Kemakmuran
Petir	: Lambang Jurusan Teknik Elektronika Industri
Sinyal	: Lambang Jurusan Teknik Komputer dan Jaringan
Kujang	: Perkakas Kabupaten Bandung dan Jawa Barat
Buku	: Pendidikan

2.1.3 Visi, Misi dan Tujuan SMKN 1 Majalaya

A. VISI

Menjadi Sekolah Menengah Kejuruan yang unggul di bidang teknologi dan berorientasi pada kepentingan daerah yang didasari Iman dan Taqwa.

Serta mampu menghasilkan lulusan yang mampu bersaing pada tingkat Regional, Nasional dan Global.

B. MISI

1. Menumbuhkan kembangkan penghayatan ajaran agama, serta menjadikan kearifan budaya lokal dan budaya bangsa sebagai wawasan dalam berfikir dan bertindak.
2. Mengembangkan semangat keunggulan dan kompetitif secara intensif kepada seluruh warga sekolah.
3. Melaksanakan kegiatan belajar mengajar secara optimal dengan berorientasi kepada pencapaian kompetensi yang berstandar Nasional

dan Internasional, dengan tetap mempertimbangkan potensi yang dimiliki peserta didik.

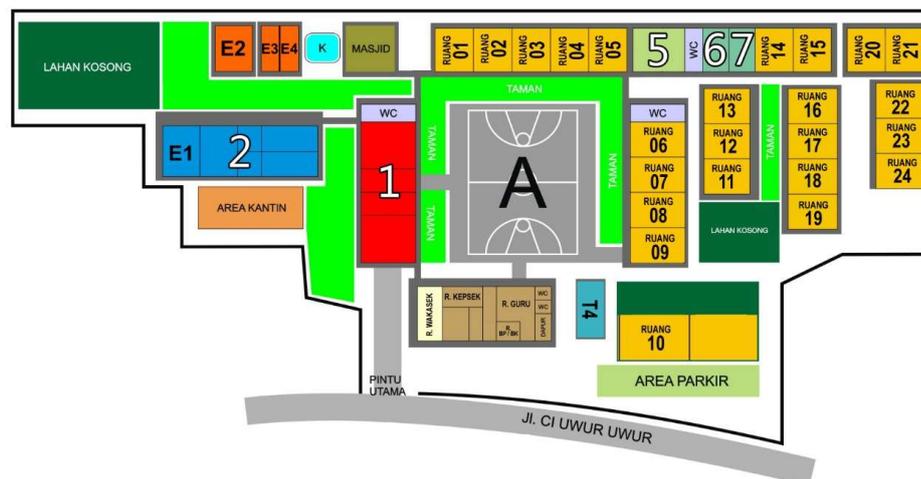
4. Mengembangkan mengintensifkan hubungan sekolah dengan dunia usaha / dunia industri dan institusi lain yang memiliki reputasi nasional dan internasional sebagai perwujudan *DEMAND DRIVEN*.

C. TUJUAN

1. Memperluas kesempatan lulusan SLTP untuk mendapat layanan pendidikan yang berkualitas.
2. Menyiapkan siswa sebagai calon tenaga kerja kelas menengah yang terampil, terdidik, dan profesional serta mampu mengembangkan diri sejalan dengan perkembangan ilmu pengetahuan dan teknologi.
3. Menyiapkan tenaga kerja sesuai dengan kebutuhan masyarakat pengguna lulusan SMK, khususnya dunia industri / usaha Kabupaten Bandung.
4. Membekali siswa dengan pengetahuan dan keterampilan sesuai dengan program keahliannya masing – masing agar dapat hidup mandiri dan berwirausaha.

2.1.4 Denah Sekolah

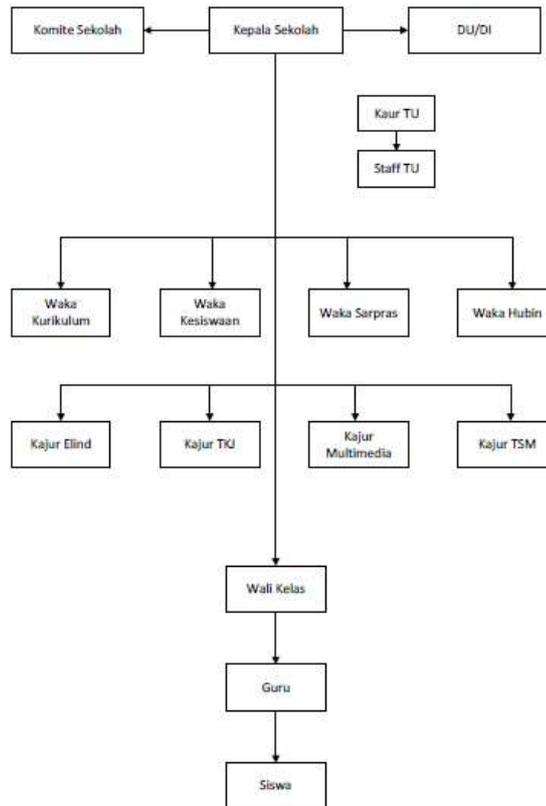
Denah sekolah SMKN 1 Majalaya dapat dilihat pada Gambar 2.2.



Gambar 2.2 Denah Sekolah

2.1.5 Struktur Organisasi

Struktur Organisasi di SMK Negeri 1 Majalaya dapat dilihat pada gambar 2.3



Gambar 2.3 Struktur Organisasi SMKN 1 Majalaya

2.1.6 Deskripsi Pekerjaan

Deskripsi pekerjaan adalah seperangkat fungsi dan tugas tanggung jawab yang dijabarkan ke dalam kegiatan pekerjaan. Tugas dan tanggung jawab dalam struktur organisasi pada Gambar 2.3. yaitu:

A. Kepala Sekolah

Kepala sekolah mempunyai tugas yang sangat berat, karena tanggung jawabnya sangat berat, karena tanggung jawabnya sangat besar, sehingga harus memahami dan menguasai manajemen dengan baik, adapun profil Kepala SMK Negeri 1 Majalaya.

B. Komite Sekolah

Komite Sekolah adalah suatu lembaga mandiri yang di bentuk dan berperan dalam peningkatan mutu pelayanan dengan memberikan pertimbangan, arah dan dukungan tenaga, sarana, dan prasarana.

C. Wakil Kepala Bidang Kurikulum

Mempunyai tugas yang berkaitan dengan komponen-komponen proses pembelajaran, yaitu program pengajaran, jadwal pelajaran, kalender pendidikan dan tugas-tugas yang terkait.

D. Wakil Kepala Bidang Kesiswaan

Tugasnya terkait dengan manajemen kesiswaan yang diawali ketika penerimaan siswa baru (PPDB), sampai dengan penyelenggaraan ujian akhir sekolah.

E. Wakil Kepala Bidang Sarana dan Prasarana

Tugasnya adalah mengelola dan memelihara segala sarana dan prasarana sekolah, serta menyusun perencanaan pengembangan sekolah.

F. Wakil Kepala Bidang Hubungan Industri

Tugas waka bidang hubin adalah mengelola dan menyampaikan informasi yang dibutuhkan oleh sekolah, serta membangun dan mengembangkan kerja sama kemitraan dengan Dunia Usaha dan Dunia Industri.

G. Guru / Tenaga Kependidikan

Tenaga profesional yang berfungsi untuk merencanakan dan melaksanakan proses pembelajaran, menilai hasil pembelajaran, melakukan pembimbingan dan pelatihan, serta melakukan penelitian dan pengabdian kepada masyarakat.

H. Tata Usaha

Melakukan koordinasi penyusunan rencana, program, dan anggaran, pengelolaan keuangan dan barang milik sekolah, evaluasi dan pelaporan

I. Wali Kelas

Wali kelas adalah guru yang membantu kepala sekolah untuk membimbing siswa dalam mewujudkan disiplin kelas, sebagai motivator untuk membangkitkan minat siswa di kelas.

2.2 Landasan Teori

Landasan teori adalah seperangkat definisi, konsep serta proposisi yang telah disusun rapi serta sistematis tentang variable-variabel dalam sebuah penelitian.

2.2.1 Otomasi

Otomasi (bahasa *Greek* berarti belajar sendiri), robotisasi atau otomasi industri atau kontrol numerik merupakan pemanfaatan sistem kontrol seperti halnya komputer yang digunakan untuk mengendalikan mesin-mesin industri dan kontrol proses untuk menggantikan operator tenaga manusia. Industrialisasi itu sendiri merupakan tahapan dalam pelaksanaan mekanisasi, dimana konsep mekanisasi tetap mesin-mesin industri dilakukan manusia sebagai operator dengan menempatkan mesin sebagai pembantunya sesuai dengan permintaan kerja secara fisik, yang jelas terjadi penurunan besar-besaran kebutuhan manusia sebagai sensor begitu juga berkaitan dengan mental kerja.[9]

Otomasi adalah teknologi yang dapat melakukan serangkaian proses secara otomatis tanpa adanya bantuan manusia. Hal ini diimplementasikan dengan menggunakan program instruksi dikombinasikan dengan sistem kontrol. Secara umum, sistem otomasi dapat didefinisikan sebagai interaksi dari mekanika, sistem kelistrikan, dan sistem komputer yang menggantikan peran manusia dalam suatu proses. [10]

2.2.2 Jaringan Komputer

Jaringan komputer adalah sekelompok komputer otonom yang saling menggunakan *protocol* komunikasi melalui media komunikasi sehingga dapat berbagi data, informasi, program aplikasi dan perangkat keras seperti *printer*, *scanner*, *CD-Drive* maupun *harddisk* serta memungkinkan komunikasi secara elektronik. Sedangkan pada aplikasi *home user*, memungkinkan komunikasi antar pengguna lebih efisien (chat), interaktif *entertainment* lebih *multimedia* (*games*, *video*, dan lain-lain).[11]

Jenis jaringan komputer bisa dikelompokkan dalam beberapa parameter yaitu berdasarkan distribusi data, jangkauan geografis, media transmisi data dan peranan dan hubungan tiap komputer dalam memproses data.[12]

A. Menurut Distribusi Data

- a. Jaringan Terpusat

Jaringan ini terdiri dari komputer klien dan *server* yang mana komputer klien yang berfungsi sebagai perantara dapat mengakses data yang disediakan oleh komputer *server*
 - b. Jaringan Terdistribusi

Merupakan perpaduan dari beberapa jaringan terpusat sehingga terdapat beberapa komputer *server* yang saling berhubungan dengan klien sehingga membentuk sistem jaringan tertentu
- B. Menurut Jangkauan Geografis.
- a. *Local Area Network* (LAN)

Merupakan jaringan yang menghubungkan 2 komputer atau lebih dalam cakupan kecil seperti laboratorium, kantor.
 - b. *Metropolitan Area Network* (MAN)

Merupakan jaringan yang mencakup satu kota besar beserta daerah setempat.
 - c. *Wide Area Network* (WAN)

Merupakan jaringan yang mencakup area geografis yang luas.
- C. Menurut Peranan Dan Hubungan Tiap Komputer Dalam Memproses Data
- a. Jaringan *Client-Server*

Pada jaringan ini terdapat 1 atau beberapa komputer *server* dan komputer klien. Komputer klien sebagai perantara untuk dapat mengakses data pada komputer *server* sedangkan komputer *server* menyediakan data yang diperlukan oleh komputer klien.
 - b. *Peer to peer*

Pada jaringan ini tidak ada komputer klien maupun komputer *server* karena semua komputer dapat melakukan pengiriman maupun penerimaan informasi sehingga semua komputer berfungsi sebagai klien sekaligus sebagai *server*.
- D. Berdasarkan Media Transmisi
- a. Jaringan Berkabel

Pada jaringan ini, untuk menghubungkan satu komputer dengan komputer lain diperlukan penghubung berupa kabel jaringan. Kabel

jaringan berfungsi untuk mengirim informasi dalam bentuk sinyal listrik antar komputer jaringan.

b. Jaringan Nirkabel

Merupakan jaringan dengan medium berupa gelombang elektromagnetik. Pada jaringan ini tidak diperlukan kabel untuk menghubungkan antar komputer karena sudah menggunakan gelombang elektromagnetik.

Topologi jaringan komputer bisa diklasifikasikan yang umumnya ada 4 jenis yaitu Topologi *Bus*, *Star*, *Ring* dan *Mesh*. [13]

A. Topologi *Bus*

Jaringan dimana beberapa *client* dihubungkan menggunakan *line* komunikasi yang terbagi yang disebut *Bus*. Jaringan *Bus* merupakan jaringan paling mudah dibuat untuk menghubungkan *client* yang ada. Namun kekurangannya adalah apabila 2 klien ingin mentransmisikan data pada saat yang sama di bus yang sama maka akan ada *collision* atau tabrakan data.

B. Topologi *Star*

Jaringan ini memiliki bentuk yang paling sederhana. Sebuah jaringan *star* terdiri atau satu *switch* atau *hub* yang berfungsi sebagai pusat untuk melakukan transmisi data di jaringan.

C. Topologi *Ring*

Topologi jaringan dimana tiap simpul akan terhubung ke 2 simpul lainnya sehingga membentuk lingkaran yang berfungsi sebagai *line* untuk transfer data. Data akan dijalankan dari simpul ke simpul yang konsekuensinya tiap simpul akan menanggapi tiap paket.

D. Topologi *Mesh*

Merupakan salah satu tipe jaringan yang bisa melakukan *recovery* sendiri dengan kata lain jaringan kan tetap bisa beroperasi walaupun ada simpul yang down atau koneksi jadi jelek.

2.2.3 IP Address

IP address merupakan 32-bit bilangan biner dimana bisa dituliskan dengan bilangan desimal dengan dibagi menjadi 4 kolom dan dipisahkan dengan titik. *IP*

address terdiri dari nomor *network* dan nomor *host*. Yang dimana nomor *network* diatur oleh suatu badan yaitu *Regional Internet Registries* (RIR).[14]

A. Representasi Alamat

Alamat *IP* versi 4 umumnya diekspresikan dalam notasi desimal bertitik (*dotted-decimal notation*), yang dibagi ke dalam empat buah oktet berukuran 8-bit. Dalam beberapa buku referensi, format bentuknya adalah w.x.y.z. Karena setiap oktet berukuran 8-bit, maka nilainya berkisar antara 0 hingga 255 (meskipun begitu, terdapat beberapa pengecualian nilai). [15] Alamat *IP* yang dimiliki oleh sebuah *host* dapat dibagi dengan menggunakan *subnet mask* jaringan ke dalam dua buah bagian, yakni:

- a. *Network Identifier/NetID* atau *Network Address* (alamat jaringan) yang digunakan khusus untuk mengidentifikasi alamat jaringan di mana *host* berada.
- b. *Host Identifier/HostID* atau *Host address* (alamat *host*) yang digunakan khusus untuk mengidentifikasi alamat *host* (dapat berupa *workstation*, *server* atau sistem lainnya yang berbasis teknologi *TCP/IP*) di dalam jaringan.

B. Pembagian Kelas Alamat *IP*

Bit pertama dari alamat *IP* memberikan spesifikasi terhadap sisa alamat dari *IP*. Selain itu juga dapat memisahkan suatu alamat *IP* dari jaringan. Alamat *network* yang sering disebut juga *NetID*, sedangkan untuk alamat *host* biasa disebut juga sebagai *HostID*. [14]

a. Kelas A

Alamat-alamat kelas A diberikan untuk jaringan skala besar. Nomor urutan *bit* tertinggi di dalam alamat *IP* kelas A selalu diset dengan nilai 0 (nol). Tujuh *bit* berikutnya—untuk melengkapi oktet pertama—akan membuat sebuah *network identifier*. 24 *bit* sisanya (atau tiga oktet terakhir) merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16.777.214 *host* tiap jaringannya. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme *Interprocess Communication* (IPC) di dalam mesin yang bersangkutan.

b. Kelas B

Alamat-alamat kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua *bit* pertama di dalam oktet pertama alamat *IP* kelas B selalu diset ke bilangan biner 10. 14 *bit* berikutnya (untuk melengkapi dua oktet pertama), akan membuat sebuah *network identifier*. 16 *bit* sisanya (dua oktet terakhir) merepresentasikan *host identifier*. Kelas B dapat memiliki 16.384 *network*, dan 65.534 *host* untuk setiap *network*-nya.

c. Kelas C

Alamat *IP* kelas C digunakan untuk jaringan berskala kecil. Tiga *bit* pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner 110. 21 *bit* selanjutnya (untuk melengkapi tiga oktet pertama) akan membentuk sebuah *network identifier*. 8 *bit* sisanya (sebagai oktet terakhir) akan merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2.097.152 buah *network*, dan 254 *host* untuk setiap *network*-nya.

d. Kelas D

Alamat *IP* kelas D disediakan hanya untuk alamat-alamat *IP multicast*, namun berbeda dengan tiga kelas di atas. Empat *bit* pertama di dalam *IP* kelas D selalu diset ke bilangan biner 1110. 28 *bit* sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*. Untuk lebih jelas mengenal alamat ini, lihat pada bagian Alamat *Multicast IPv4*.

e. Kelas E

Alamat *IP* kelas E disediakan sebagai alamat yang bersifat "eksperimental" atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat *bit* pertama selalu diset kepada bilangan biner 1111. 28 *bit* sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*.

2.2.4 *SSH*

SSH merupakan protokol standar yang membentuk jalur yang aman pada komunikasi antar komputer. *SSH* menggunakan teknik enkripsi *public key* pada

sistem autentikasi pengguna untuk mengakses komputer lain. *SSH* memberikan sistem enkripsi pada jalur yang digunakan, sehingga memberikan tingkat keamanan data yang tinggi.[16]

SSH biasa digunakan untuk melakukan *remote login* dan memberikan perintah ke komputer *remote*, tetapi dapat digunakan juga sebagai *tunnel* jaringan, transfer *file* melalui protokol *SFTP* dan *SCP*.[14]

2.2.5 DHCP

DHCP (Dynamic Host Controller Protocol) adalah protokol pengalamatan *host* secara dinamis. Dalam sebuah jaringan yang besar, akan ada bagian yang pengalamatan *IP address* tidak begitu kritis. Di bagian ini pengalamatan *IP* bisa dilakukan secara dinamis dan otomatis.[17] Terdapat 2 aktor pada *DHCP* yaitu *DHCP Server* dan *DHCP Client*.

DHCP Server adalah *server* yang melayani protokol untuk penyewaan *IP* sedangkan *DHCP Client* adalah komputer yang meminta *IP Address*.

Terdapat 5 tahapan yang dilakukan oleh *DHCP* untuk memberikan konfigurasi *IP Address*, antara lain :

A. IP Lease Request

Merupakan proses dimana *client* meminta *IP Address* ke *server*

B. IP Lease Offer

Tahapan dimana *DHCP Server* memberikan *IP Address* yang telah diminta oleh *DHCP Client*.

C. IP Lease Selection

Client memilih penawaran *DHCP server* yang pertama diterima dan kembali melakukan *broadcast* dengan *message* menyetujui peminjaman tersebut kepada *DHCP Server*.

D. IP Lease Acknowledge

DHCP server memberikan jawaban atas pesan tersebut berupa konfirmasi *IP Address* dan informasi lain kepada *client* dengan sebuah *Acknowledgment*. Kemudian *client* melakukan inisialisasi dengan mengikat (*binding*) *IP Address* tersebut dan *client* dapat bekerja pada jaringan tersebut.

E. Lease Period

Setelah periode waktu tertentu, maka pemakaian *DHCP client* tersebut dinyatakan selesai dan *client* tidak memperbaharui permintaan kembali, maka *IP Address* tersebut dikembalikan kepada *DHCP server*, dan *server* dapat memberikan *IP Address* tersebut kepada *client* yang membutuhkan.

2.2.6 DNS Server

DNS adalah sebuah sistem yang menyimpan informasi tentang nama *host* ataupun nama *domain* dalam bentuk basis data tersebar (*distributed database*) di dalam jaringan komputer.[17] *DNS* menyediakan alamat *IP* untuk setiap nama *host* dan mendata setiap *server* transmisi surat (*mail exchange server*) yang menerima surel (*email*) untuk setiap *domain*.

DNS dibagi menjadi beberapa jenis yaitu sebagai *primary (master)*, *secondary (slave)* dan *cache*.

a. Primary (Master)

Primary atau *master* adalah pemegang daftar lengkap dari sebuah *domain* yang dikelolanya. *Server* ini memegang otoritas penuh atas *domainnya*. Otoritas penuh di sini berarti *server* ini yang bertanggung jawab untuk ditanyai nama-nama *host* ber-*domain* yang dikelola dan *sub-sub domain* dibawahnya.

b. Secondary (Slave)

Digunakan untuk menjawab permintaan dari *name server* lain. *Server* slave merupakan *backup* dari *server master*. *Server* ini mendapatkan informasi *name space* dari *name server master*. *Server master* akan mengirim perubahan tersebut ke *slavenya* setiap periode tertentu.

c. Cache

Jenis ini tidak mempunyai data nama-nama *host* dari *domain* tertentu. Ia hanya mencari jawaban dari beberapa *DNS server* terdekat. Setelah jawaban didapatkan, datanya disimpan dalam *cache* untuk keperluan mendatang. *DNS server cache* merupakan yang paling mudah untuk dikonfigurasi.

2.2.7 *NTP Client*

Network Time Protocol (NTP) merupakan sebuah mekanisme atau protokol yang digunakan untuk melakukan sinkronisasi terhadap penunjuk waktu dalam sebuah sistem komputer dan jaringan. Proses sinkronisasi ini dilakukan di dalam jalur komunikasi data yang biasanya menggunakan protokol komunikasi *TCP/IP*. Sehingga proses ini sendiri dapat dilihat sebagai proses komunikasi data biasa yang hanya melakukan pertukaran paket-paket data saja.[18]

2.2.8 *Firewall*

Firewall merupakan suatu *tools* yang di dalam sistem operasi yang berfungsi sebagai alat untuk melakukan penyaringan terhadap lalu lintas data. *Firewall* dapat digunakan untuk melakukan seleksi terhadap paket-paket yang datang baik *input*, *output* maupun *forward* berdasarkan *IP address*, identitas jaringan, *port*, *source*, *destination*, protokol yang digunakan bahkan berdasarkan tipe koneksi terhadap setiap paket (data yang diinginkan).[17]

2.2.9 *Routing*

Routing adalah suatu protokol yang digunakan untuk mendapatkan rute dari satu jaringan ke jaringan yang lain. Rute ini, disebut dengan *route* dan informasi *route* secara dinamis dapat diberikan ke *router* yang lain ataupun dapat diberikan secara statis ke *router* lain.[13]

Ada dua macam *routing* yang biasa dilakukan yaitu *routing* statis dan *routing* dinamis.[14]

A. *Routing* Statis

Melakukan konfigurasi *routing* secara manual sehingga akan mengalami kesulitan bila harus memasukan atau menghapus tabel *route* statis dan *routing* ini dibangun berdasarkan definisi dari *administrator*.

B. *Routing* Dinamis

Algoritma ini dapat membuat perangkat *router* untuk dapat menentukan jalur *routing*-nya secara otomatis, dengan cara menjelajah jaringan tersebut dan bertukar informasi *routing* antar *router*. Terdapat 3 kategori tentang algoritma dinamik, yaitu :

- a. *Distance Vector*
- b. *Link State*

c. *Hybrid*

2.2.10 *OSPF*

Open Shortest Path First (OSPF) adalah salah satu jenis *routing*, lebih baik, lebih kuat, lebih cepat dari pendahulunya. *OSPF* menggunakan protokol *routing* interior dengan algoritma *link-state* yang di rancang untuk melampaui kemampuan dari keterbatasan *distance vector routing*. Bertepatan dengan bertambahnya *router* di sebuah area akan bertambah pula informasi yang harus dimiliki oleh *router* pada area yang sama dalam waktu yang sama. maka *OSPF* lah yang paling tepat untuk proses *routing* tersebut. [19]

2.2.11 *Hotspot*

Hotspot adalah salah satu bentuk pemanfaatan teknologi *Wireless LAN* pada lokasi-lokasi *public* seperti perpustakaan, taman area kampus dan lain-lain. *User* dapat mengakses *hotspot* melalui *mobile computer* tanpa menggunakan koneksi kabel dengan tujuan dapat mengakses suatu jaringan seperti *internet*. [20]

2.2.12 *Mikrotik*

Mikrotik merupakan produsen *software* dan *hardware router Mikrotik*. *Mikrotik* merupakan sistem operasi *Linux base* yang diperuntukkan sebagai *network router*. Didesain untuk memberikan kemudahan bagi penggunaannya. [21]

Jenis-Jenis *Mikrotik* yang dikenal yaitu *Mikrotik RouterOS* dan *Mikrotik RouterBOARD*.

a. *Mikrotik RouterOS*

Mikrotik RouterOS yang berbentuk *software*. Dapat diinstal pada komputer rumahan (PC). *Mikrotik RouterOS* adalah sebuah *software* yang berfungsi mengubah PC (komputer) menjadi sebuah *router*.

b. *Mikrotik RouterBOARD*

Mikrotik RouterBoard yaitu *Hardware router* diproduksi oleh *mikrotik* yang sudah terdapat *mikrotik RouterOS* sehingga pengguna tidak harus menginstall *RouterOS* ke dalam PC .

2.2.13 *Bahasa Pemrograman Python*

Python merupakan bahasa pemrograman dinamis yang mendukung pemrograman berbasis objek. *Python* didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Namun pada prinsipnya *Python* dapat diperoleh dan

dipergunakan secara bebas, bahkan untuk kepentingan komersial. Karena lisensi *Python* tidak bertentangan baik menurut definisi *Open Source* maupun *General Public License (GPL)*. [22]

2.2.14 Paramiko

Paramiko adalah implementasi *Python* (2,7, 3,4+) dari protokol *SSHv2*, menyediakan fungsionalitas klien dan *server*. Meskipun memanfaatkan ekstensi *Python C* untuk kriptografi tingkat rendah (*Kriptografi*), *Paramiko* sendiri adalah antarmuka *Python* murni di sekitar konsep jaringan *SSH*. [23]

2.2.15 Website

Website adalah salah satu aplikasi yang berisikan dokumen-dokumen *multimedia* (*text*, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol *http* (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*. [24]

2.2.16 MySQL

MySQL adalah sebuah *database management system* (DBMS) populer yang memiliki fungsi sebagai *relational database management system* (RDBMS). Selain itu *MySQL software* merupakan suatu aplikasi yang sifatnya *open source* serta *server* basis data *MySQL* memiliki kinerja sangat cepat, reliable, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems*. [25]

2.2.17 Object Oriented Programming

Object Oriented Programming adalah paradigma pemrograman yang memandang perangkat lunak sebagai kumpulan objek yang saling berinteraksi di dalam suatu sistem. [26]

Pemrograman berorientasi objek (*Object Oriented Programming*) menekankan konsep sebagai berikut:

1. *Class* merupakan sekumpulan objek yang memiliki atribut-atribut dan *method*. *Class* merupakan penggambaran dari sebuah objek atau sifat objek dan juga apa yang bisa dilakukan oleh objek tersebut. Ada juga yang disebut dengan *superclass*, sebuah *class* induk yang nantinya mempunyai *class-class* yang terdiri dari *class* dan *subclass*.

2. Objek dalam *OOP* adalah sebuah suatu bentuk dari *class*. Objek menggambarkan kemampuan untuk menyimpan informasi mengenai hal tersebut, berinteraksi dengan hal tersebut atau keduanya.
3. *Encapsulation*/pengkapsulan, yang merupakan pembatasan ruang lingkup program terhadap data yang diproses supaya data terlindungi oleh prosedur atau objek lain, kecuali prosedur yang berada di objek itu sendiri.
4. *Polymorphism* adalah konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku yang berbeda, bahwa operasi yang sama mungkin memiliki perbedaan dalam *class* yang berbeda.
5. *Inheritance* (pewarisan), yaitu kepemilikan yang bersifat implisit dari fitur *subclass* yang didefinisikan dalam *superclass*. Fitur tersebut mencakup *variables* dan *method*.

2.2.18 UML

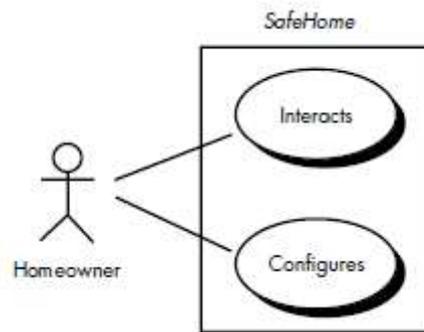
UML (Unified Modeling Language) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual.[26]

2.2.19 Use Case Diagram

Use Case Diagram adalah *diagram UML* yang berguna untuk memberikan penjelasan mengenai fungsi suatu sistem yang sedang dikembangkan kepada *user* menggunakan istilah-istilah yang mudah dimengerti.[27] *Use Case Diagram* memberikan representasi grafis mengenai urutan aktivitas berupa interaksi yang terjadi antara *user* (didalam *Use Case Diagram* disebut dengan *Actor*) dan *system* serta tujuan dari penggunaan *system* tersebut.

2.2.20 Use Case

Pemodelan *Use-case* mengidentifikasi dan mendeskripsikan fungsi-fungsi pada *system* dengan alat yang disebut *use cases*. Sebuah *use case* merepresentasikan sebuah tujuan dari sistem dengan menggambarkan urutan aktivitas dan interaksi *user* dalam mencapai tujuan. *Use case* digambarkan dengan sebuah bentuk elips dengan nama dari *use case* yang muncul dari atas, bawah, atau dalam *elips*.[22] Contoh *use case* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Contoh *Use Case*[22]

2.2.21 Actor

Use Case diinisiasi atau dipicu oleh pengguna *external* yang disebut *Actor*. *Actor* merupakan segala sesuatu yang berinteraksi langsung dengan sistem seperti orang, benda atau lainnya. [26]

2.2.22 *Use Case Scenario*

Use Case scenario merupakan penjelasan secara tekstual dari sekumpulan skenario interaksi. Setiap skenario mendeskripsikan urutan aksi/langkah yang dilakukan *aktor* ketika berinteraksi dengan sistem, baik yang berhasil maupun gagal. [27]

Use Case scenario dijelaskan secara tekstual dalam beberapa format tergantung kebutuhannya, yaitu singkat (*brief*), informal (*casual*), atau lengkap (*fully dressed*) [28], yang bisa dijelaskan dalam bentuk tabel dengan 1 kolom atau 2 kolom [29]. Pada format singkat, penjelasan diberikan cukup 1 paragraf yang mengacu hanya pada skenario yang berhasil. Pada format informal, penjelasan diberikan dalam beberapa paragraf yang mencakup semua skenario, baik yang berhasil maupun gagal. Sedangkan, pada format lengkap, penjelasan dibuat secara detil disertai dengan bagian-bagian pendukung yang penting. Format terakhir ini yang banyak digunakan di dalam praktik. Bagian-bagian penting tersebut adalah [28]:

- A. **Aktor primer** (*primary actor*), yaitu aktor yang menginisiasi layanan sistem untuk mencapai tujuan dari aktor tersebut. Jumlah aktor primer dimungkinkan lebih dari 1.
- B. **Prakondisi** (*preconditions*), yaitu kondisi spesifik yang harus terpenuhi sebelum sebuah *Use Case* bisa diinisiasi atau dieksekusi oleh aktor primer. Jumlah prakondisi bisa lebih dari 1 keadaan.

- C. **Alur utama** (*main or basic flow*), yaitu jalur interaksi yang mengarahkan pada skenario yang berhasil sehingga tujuan aktor bisa terpenuhi. Jalur ini hanya terdiri dari 1 jalur saja.
- D. **Alur alternatif** (*alternative flows*), yaitu jalur alternatif dari interaksi yang terjadi antar aktor dengan sistem yang mencakup pencabangan (pilihan) maupun skenario yang gagal sehingga tujuan aktor tidak terpenuhi. Jalur ini bisa terdiri dari lebih dari 1 jalur kemungkinan.
- E. **Kondisi akhir** (*postconditions*), yaitu kondisi spesifik yang harus terjadi ketika *Use Case* berhasil dijalankan atau dieksekusi secara lengkap, sebagai representasi dari tujuan yang ingin dicapai oleh aktor primer. Jumlah kondisi akhir bisa lebih dari 1 keadaan.

2.2.23 Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur urutan yang ada pada sebuah sistem dan menjelaskan alur dari sebuah *use case*. [26]

2.2.24 Class Diagram

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. [26]

Class memiliki tiga area pokok :

- A. Nama (dan *stereotype*)
- B. Atribut
- C. Metoda

2.2.25 Sequence Diagram

Sequence diagram adalah *diagram* UML yang menggambarkan tahap-tahap interaksi yang dilakukan objek berdasarkan urutan waktu. [26]

2.2.26 Application Programming Interfaces

APIs dikembangkan karena adanya tren industri yang baru, seperti:

- 1.) *Distributed System* untuk menyediakan layanan yang efisien, meningkatkan *reliability* dan *availability*, dan kelebihan lain untuk integrasi sistem

- 2.) Teknologi perangkat lunak, seperti pemrograman berbasis objek untuk menyediakan layanan terfokus pada bagian bagian yang penting, penyembunyian data, dan abstraksi untuk meningkatkan *software extensibility* dan penggunaan kembali;
- 3.) Jaringan yang terbagi, yang dapat digantikan pada pemrosesan yang terpusat melalui jaringan;
- 4.) Komponen teknologi perangkat keras dan perangkat lunak untuk meningkatkan *reliability* dan stamina untuk beberapa poin di mana dapat meningkatkan penggabungan jaringan komunikasi yang menunjang aplikasi *high-reliability*. [30]

2.2.27 Javascript Object Notation (Json)

Javascript Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis, serta mudah diterjemahkan dan dibuat (*generate*). Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*. *JSON* merupakan format teks yang tidak tergantung pada bahasa pemrograman apapun karena menggunakan bahasa yang umum digunakan oleh programmer keluarga *C* antara lain *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python*, dll. Oleh karena itu sifat-sifat tersebut, JSON ideal sebagai bahasa pertukaran data. [31]