

BAB II

TEORI PENUNJANG

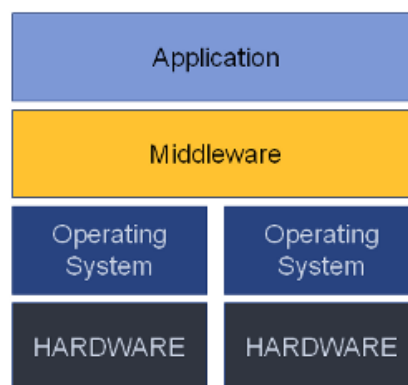
Pada bab ini akan membahas tentang teori dan komponen penunjang yang akan digunakan pada perancangan *RFID Middleware* yang mengirimkan data dari *reader* ke aplikasi *Smart Warehouse* berbasis Android menggunakan media transmisi *wireless* dengan jaringan lokal.

2.1 Smart Warehouse

Smart Warehouse adalah sistem gudang cerdas yang melakukan terjadinya keluar masuk barang disuatu pergudangan yang terintegrasi dengan Internet, sehingga memudahkan admin gudang dalam melakukan aktivitas pergudangan. *Smart Warehouse* dapat mengintegrasikan kegiatan pergudangan admin gudang secara efisien dan ekonomis[1].

2.2 Middleware

Dalam arsitektur komputer, *Middleware* adalah perangkat lunak yang membuat suatu jaringan pertukaran informasi antar aplikasi komputer yang berbeda. Jaringan diimplementasikan dengan penggunaan teknik pertukaran informasi yang sama pada semua aplikasi yang terimplikasi dengan bantuan perangkat lunak terkomposisi. Gambar 2.1 merupakan konsep *Middleware* secara umum[2].



Gambar 2.1 Konsep Middleware

Gambar 2.1 merupakan konsep perangkat lunak *Middleware* yang menjamin komunikasi antara aplikasi komputer yang terimplikasi apapun karakteristik

hardware maupun software dalam jaringan komputer, protokol jaringan dan sistem operasi yang terimplikasi.

2.3 RFID Middleware

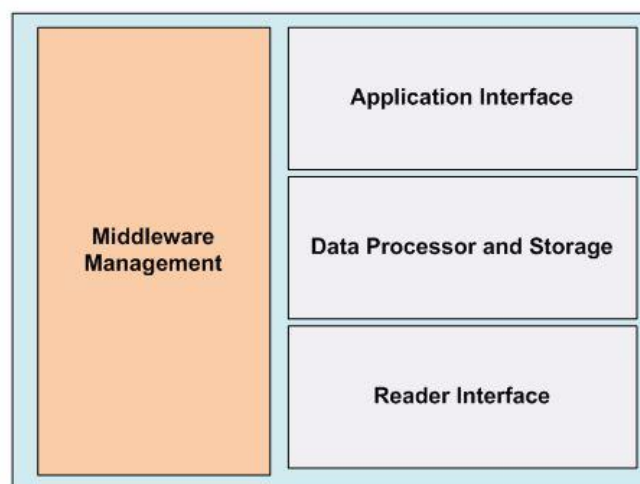
RFID *Middleware* menggabungkan perangkat identifikasi frekuensi radio (RFID), yang berada diantara reader RFID dan aplikasi perusahaan. *Middleware* memiliki beberapa fungsi dan memainkan peran utama dalam operasi sistem RFID dan manajemen. *Middleware* tidak hanya mengelola reader RFID saja dapat berkomunikasi antara perangkat yang membuat suatu jaringan pertukaran informasi antara aplikasi komputer yang berbeda. Beberapa perangkat lunak *middleware* menjamin komunikasi antara aplikasi apapun komputer yang terimplikasi dan apapun karakteristik perangkat keras maupun perangkat lunak dalam jaringan komputer, protokol jaringan, dan sistem operasi yang terimplikasi. RFID *Middleware* ini mengatur arus data dari beberapa *reader* dan menyaring data yang tidak penting[3].

Sebuah RFID *Middleware* merupakan antarmuka yang berada di antara perangkat keras RFID dan aplikasi RFID. Ini memberikan keuntungan sebagai berikut :

1. Menyembunyikan rincian perangkat keras dari aplikasi.
2. Menangani dan memproses data RFID mentah sebelum diteruskan ke aplikasi.
3. Menyediakan antarmuka tingkat aplikasi untuk mengelola *reader* RFID dan *query* data RFID.

2.3.1 Lapisan utama RFID *Middleware*

Berikut ini merupakan empat komponen utama lapisan RFID *Middleware*.



Gambar 2.2 Komponen RFID *Middleware*

Pada bagian ini akan menjelaskan setiap bagian dari lapisan utama dari RFID *Middleware* :

1. Antarmuka *Reader*

Pembaca antarmuka merupakan lapisan terendah dari RFID *middleware* yang menangani interaksi dengan hardware RFID. Bagian ini memelihara driver perangkat dari semua perangkat yang didukung oleh sistem, dan mengelola semua parameter perangkat keras yang terkait seperti protokol *reader*, dan komunikasi *host-side*.

2. Pengolahan Data dan Penyimpanan

Pengolahan data dan penyimpanan pada bagian ini bertanggung jawab untuk memproses dan menyimpan data mentah yang berasal dari reader RFID. Bagian ini juga memproses data level yang terkait dengan aplikasi tertentu.

3. Antarmuka Aplikasi

Antarmuka aplikasi menyediakan aplikasi dengan API untuk mengakses, berkomunikasi, dan mengkonfigurasi RFID *middleware*. Bagian ini mengintegrasikan aplikasi *Smart Warehouse* dengan RFID *middleware* dengan menerjemahkan permintaan aplikasi pada *middleware* tingkat rendah.

4. Manajemen *Middleware*

Manajemen *middleware* membantu mengelola konfigurasi RFID *middleware*, dan menyediakan kemampuan sebagai berikut :

- Menambah, mengkonfigurasi, dan memodifikasi pembaca RFID yang terhubung.
- Mengubah parameter tingkat aplikasi seperti filter dan penghilangan duplikasi waktu.
- Menambah dan menghapus layanan yang didukung oleh RFID *middleware*.

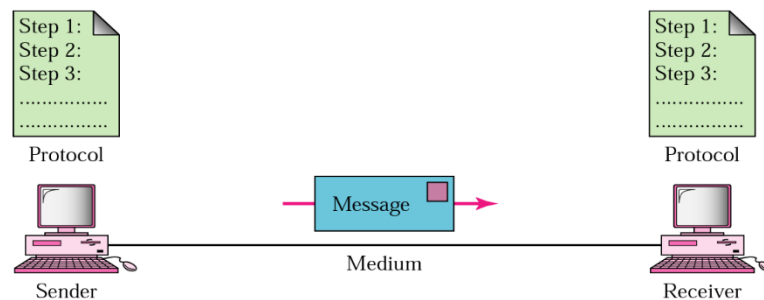
Reader RFID biasanya diabstraksikan sebagai *reader* logis yang baik dari beberapa *reader*. Mekanisme pengelompokan ini digunakan dimana ada kebutuhan untuk menangkap data dari tempat tertentu seperti gudang.

2.4 Komunikasi Data

Komunikasi data merupakan pertukaran data antara dua perangkat melalui beberapa bentuk media transmisi seperti kabel kawat. Guna melangsungkan komunikasi data, perangkat yang berkomunikasi haruslah menjadi bagian dari sistem komunikasi yang terdiri dari kombinasi perangkat keras (*hardware*) dan perangkat lunak (*software*)[4]. Keefektifan suatu sistem komunikasi data adalah tergantung dari empat sifat mendasar, yaitu sebagai berikut:

1. Pengiriman. Sistem harus mengirimkan data ke tujuan yang diinginkan. Data yang dikirimkan tersebut wajib diterima oleh perangkat yang dituju atau *user* yang dituju.
2. Akurasi. Sistem harus mengirimkan data yang akurat. Perubahan data yang terjadi dalam transmisi akan menyebabkan data menjadi tidak dapat digunakan.
3. Ketepatan waktu. Sistem harus mengirimkan data pada waktu yang tepat. Data yang terlambat dikirimkan tidak dapat digunakan.
4. *Jitter*. *Jitter* mengacu pada variasi waktu kedatangan paket. Biasanya terjadi *delay* acak pada pengiriman paket suara atau video.

Sistem komunikasi data memiliki 5 komponen yang ditunjukkan pada Gambar 2.3



Gambar 2.3 Lima Komponen Komunikasi Data

1. Pesan (*Message*). Pesan adalah informasi (data) yang akan dikomunikasikan.
2. Pengirim (*Sender*). Pengirim adalah perangkat yang mengirimkan data pesan.
3. Penerima (*Receiver*). Penerima adalah perangkat yang menerima data pesan.
4. Media transmisi (*Medium*). Media transmisi adalah media yang digunakan untuk melakukan pengiriman data.

5. Protokol (*Protocol*). Protokol adalah aturan-aturan yang berfungsi untuk mengatur komunikasi data.

2.5 Media Transmisi

Berdasarkan media transmisi, jaringan komputer dibagi menjadi :

1. Media Transmisi Kabel (*Wired*) Kabel merupakan media penghubung yang paling banyak digunakan pada jaringan LAN. Beberapa kabel yang digunakan dalam membangun jaringan diantaranya, kabel twisted pair, kabel coaxial dan kabel fiber optic. Kabel twisted pair dibagi menjadi dua yaitu, unshielded twisted pair dan shielded twisted pair.
2. Media Transmisi Tanpa Kabel (*Wireless*) Media transmisi wireless ini merupakan media transmisi tanpa kabel yang masing-masing perangkatnya terhubung menggunakan transmisi gelombang radio, microwave atau infrared. Beberapa perangkat untuk penunjang media transmisi ini yaitu switch, hub, router dan access point[5].

2.6 Protokol

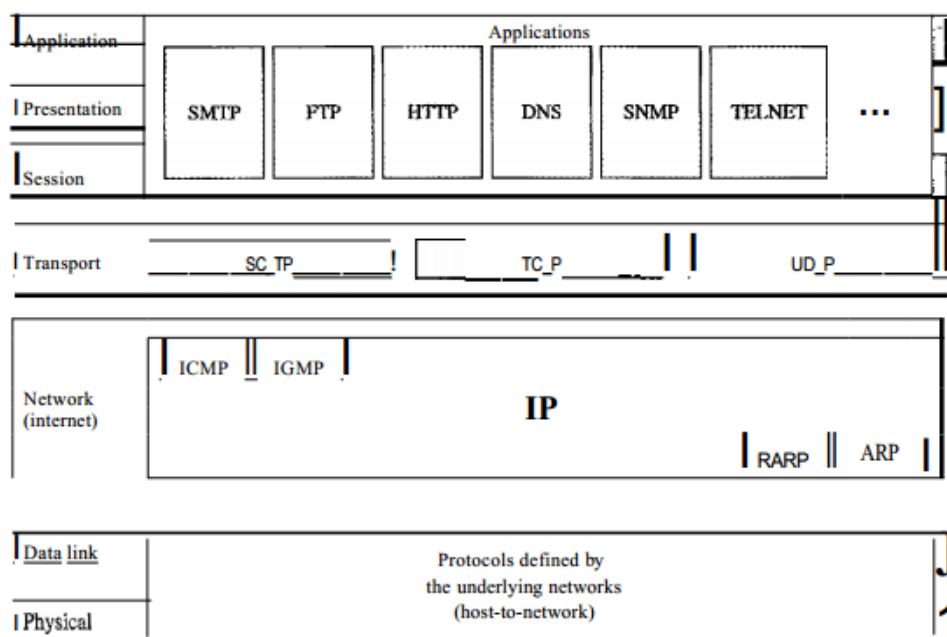
Apabila ada dua buah sistem saling berkomunikasi dengan bahasa yang berlainan, tentunya dua sistem tersebut tidak akan saling memahami. Untuk itu sistem membutuhkan sebuah mekanisme pengaturan bahasa yang dapat dipahami oleh dua sistem tersebut, sehingga pertukaran informasi antar sistem akan dapat terjadi dengan benar[6].

Protokol merupakan sarana komunikasi antara mesin melalui jaringan yang terstandarisasi. Protokol mengizinkan data untuk ambil bagian dalam transmisi kilat, kemudia ditransmisikan, lalu dikumpulan kembali sesuai arah dengan perintah yang benar.

2.6.1 Protokol TCP/IP

TCP/IP protokol suite dikembangkan sebelum OSI model. Oleh karena itu, lapisan dalam protokol TCP/IP tidak persis sesuai dalam OSI model. Protokol TCP/IP asli didefinisikan sebagai memiliki empat lapisan: host-untuk-jaringan, internet, transportasi, dan aplikasi. Namun, bila TCP/IP dibandingkan OSI, kita dapat mengatakan bahwa lapisan host-untuk-jaringan setara dengan kombinasi fisik

dan lapisan data link. Lapisan internet setara dengan lapisan jaringan, dan lapisan aplikasi kira-kira adalah melakukan pekerjaan sesi, presentasi, dan aplikasi lapisan dengan lapisan transport di TCP/IP merawat bagian dari tugas-tugas lapisan sesi. Sehingga TCP/IP protokol *suite* terdiri dari lima lapisan: fisik, data link, Jaringan, transportasi, dan aplikasi. Empat lapisan memberikan standar fisik, antarmuka jaringan, internetworking, dan fungsi transportasi yang sesuai dengan empat lapisan dari OSI model. Tiga lapisan paling atas dalam OSI model, namun, diwakili di TCP/IP oleh lapisan tunggal yang disebut lapisan aplikasi.



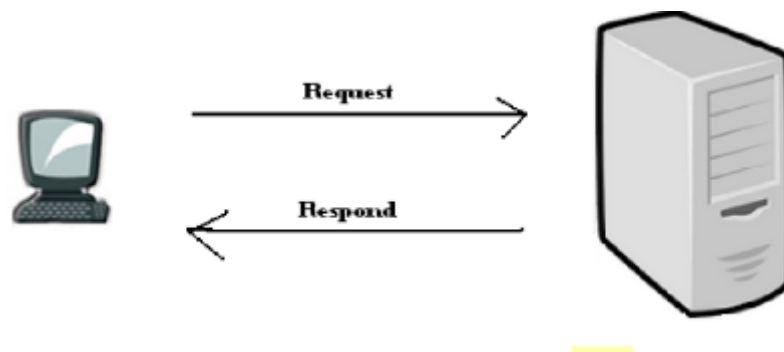
Gambar 2.4 TCP/IP data model OSI

Dari gambar 2.4 dijelaskan bahwa TCP/IP merupakan sebuah protokol hirarkis terdiri dari modul yang interaktif, masing-masing yang menyediakan fungsionalitas tertentu, namun, modul tidak perlu saling bergantung. Sedangkan OSI model menetapkan fungsi yang milik masing-masing lapisan, lapisan protokol TCP/IP mengandung relatif independen protokol yang dapat dicampur dan dicocokkan tergantung pada kebutuhan sistem. Istilah hirarkis berarti bahwa setiap tingkat tinggi protokol ini didukung oleh satu atau lebih protokol tingkat rendah. Pada transport layer ini, TCP/IP mendefinisikan tiga protokol yaitu *Transmission Control Protocol* (TCP), *User Datagram Protocol* (UDP) dan *Stream Control*

Transmission Protocol (SCTP). Lapisan jaringan, protokol utama yang didefinisikan oleh TCP/IP adalah *Internetworking Protocol* (IP). Ada juga beberapa protokol lain yang mendukung pergerakan data dalam lapisan ini.

2.6.2 HTTP (*Hypertext Transfer Protocol*)

HTTP merupakan standar protokol yang digunakan untuk mengakses halaman web. HTTP versi awal yaitu HTTP versi 1.0 didefinisikan dengan RFC 1945 dan HTTP versi 1.1 didefinisikan dengan RFC 2068. HTTP menggunakan port 80 untuk melakukan komunikasi. Client akan memberikan request ke server HTTP dan server akan memberikan respon sesuai dengan permintaan dari Client. Komunikasi ini menggunakan model Client-Server.



Gambar 2.5 Request dan respon HTTP

Pesan Request dan format status untuk pesan HTTP memiliki format tersendiri, seperti pada gambar 2.5. Tipe request dan status code HTTP pun bermacam-macam.

2.7 Web Server

Web server merupakan perangkat lunak yang memberikan layanan data, berfungsi menerima permintaan HTTP atau HTTPS (*HyperText Transfer Protocol Secure*) dari *client* yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML, konsep *web server* yaitu *Web server* merupakan mesin aplikasi atau perangkat lunak yang beroperasi dalam mendistribusikan halaman *web* ke *user*, tentu saja sesuai dengan permintaan *user*.

Hubungan antara *web server* dan *browser* merupakan gabungan atau jaringan komputer yang berada di seluruh dunia. Setelah terhubung secara fisik, protokol

TCP/IP yang memungkinkan semua komputer dapat berkomunikasi antar satu dengan lainnya. Pada saat aplikasi *browser* meminta data halaman *web* ke *server* maka instruksi permintaan data oleh *browser* tersebut dikemas dalam TCP yang merupakan protokol pengiriman dan dikirim ke alamat yang merupakan protokol berikutnya yaitu HTTP. Data yang di-*parsing* dari *browser* ke *web server* disebut sebagai HTTP *request* yang meminta halaman *web* dan kemudian *web server* akan mencari data HTML yang dibutuhkan dan dikemas dalam protokol TCP kemudian dikirim kembali ke *browser*. Data yang dikirim dari *server* ke *browser* disebut sebagai HTTP *response*. Jika data yang diminta oleh *browser* tidak ditemukan pada *web server* maka akan menampilkan *error* pada halaman *web*[7]

2.8 Web Service

Web services adalah sebuah sistem perangkat lunak yang didesain untuk mendukung interaksi yang interoperable antar mesin melalui sebuah jaringan. Walaupun konsep-konsep yang membentuk SOA telah ada sebelum web services muncul, web services memiliki peran penting didalam SOA. Interoperabilitas merupakan salah satu manfaat utama yang diperoleh dari penerapan layanan web. Hal ini dikarenakan web services dibangun diatas protokol-protokol yang sudah terkenal dan memiliki platform yang independent, seperti HTTP, XML, UDDI, dan WSDL[8].

2.9 Router

Router adalah sebuah alat yang digunakan untuk mengatur ruter sinyal atau data yang ada di jaringan komputer sehingga dapat diarahkan menuju ke rute tertentu yang telah diatur sebelumnya dan menghasilkan suatu hubungan antar jaringan komputer itu sendiri[9]. Fungsi sebuah router dapat memeriksa setiap paket yang datang kepadanya dan menentukan apakah paket tersebut ditujukan ke jaringan IP lokalnya atau ke sebuah jaringan di luar wilayah router (*remote*).

2.10 Pemrograman Terstruktur

Pemrograman Terstruktur adalah suatu proses untuk mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program. Selain pengertian diatas Pemrograman Terstruktur adalah suatu aktifitas pemrograman dengan memperhatikan urutan langkah-langkah perintah secara sistematis, logis ,

dan tersusun berdasarkan algoritma yang sederhana dan mudah dipahami. Prinsip dari pemrograman terstruktur adalah Jika suatu proses telah sampai pada suatu titik / langkah tertentu , maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya / kembali lagi ke baris sebelumnya, kecuali pada langkah – langkah untuk proses berulang (Loop)[10]. Berikut Sifat-sifat pemrograman terstruktur :

1. Memuat teknik pemecahan masalah yang logis dan sistematis
2. Memuat algoritma yang efisien, efektif dan sederhana
3. Program disusun dengan logika yang mudah dipahami
4. Tidak menggunakan perintah *GOTO*
5. Biaya pengujian program relatif rendah
6. Memiliki dokumentasi yang baik
7. Biaya perawatan dan dokumentasi yang dibutuhkan relatif rendah.

2.11 Bahasa Pemrograman PHP

PHP, singkatan rekursif dari *PHP: Hypertext Preprocessor*, adalah bahasa pemrograman yang dapat digunakan untuk tujuan umum, sama seperti bahasa pemrograman lain: C, C++, Pascal, Python, Perl, Ruby dan sebagainya. Meskipun PHP lebih populer digunakan untuk pengembangan aplikasi *web*. Meskipun pengembangan PHP lebih difokuskan untuk proses pembuatan aplikasi *web* (sering disebut: *server-side scripting*), tapi sebenarnya PHP memiliki kemampuan lebih dari itu[11]. PHP dapat digunakan untuk membuat tiga tipe aplikasi, yaitu :

1. Aplikasi *web* (*server-side-scripting*).
2. Program CLI (*command-line scripting*)
3. Aplikasi *desktop* (GUI).

2.12 Basis Data

Basis Data adalah kumpulan data yang saling berelasi . Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan, karakter, atau symbol)[12]. Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.

2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (redudancy) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan.

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali.

2.12.1 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau dikenal dengan DBMS(*database management system*), *database* ini *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU(*General Public License*)[13]. MySQL memiliki beberapa kelebihan dan keuntungan dibanding *database* lain, di antaranya adalah :

1. MySQL merupakan ssistem manajemen basis data *open source*, yaitu *software* ini bersifat *free* atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
2. MySQL mempunyai performa yang tinggi.
3. MySQL dapat diakses melalui protokol ODBC(*Open Database Connectivity*) buatan Microsoft. Ini menyebabkan MySQL dapat diakses oleh banyak *software*.
4. Semua klien dapat mengakses server dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses *database*.
5. MySQL merupakan *database* yang mampu menyimpan data berkapasitas besar, sampai berukuran Gigabyte.
6. MySQL dapat berjalan diberbagai sistem operasi seperti Linux, Windows, Solaris, dan lain-lain.