BAB II
TEORI PENUNJANG

Pada bab ini membahas tentang teori serta software yang akan digunakan pada sistem *Smart Warehouse* yang menggunakan jaringan lokal. Pembahasan mencakup *Smart Warehouse,* Android, *Object Oriented Programming*, *Unifield Modeling Language*, *Java* dan SQLite.

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 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *Middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi pada pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance,* konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia[2].

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. Tidak hanya menjadi sistem operasi di *smartphone,* saat ini Android menjadi pesaing utama dari Apple pada sistem operasi Table PC. Pesatnya pertumbuhan Android dikarenakan Android itu sendiri adalah *platform* yang sangat lengka baik itu sistem operasinya, Aplikasi dan *Tool* Pengembangan, Market aplikasi Android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga

Android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

Saat ini Android telah melalui cukup banyak pembaruan sistem operasi sejak pertama dirilis. Tabel 2.1 menunjukkan berbagai versi Android beserta dengan tanggal rilis, nama kode dan level API (*Application Programming Interface*).

Tabel 2.1 Versi Android

Versi Android	**Tanggal Rilis**	**Nama Kode**	**Level API**
1.0	23 September 2008	-	1
1.1	9 Februari 2009	-	2
1.5	30 April 2009	Cupcake	3
1.6	15 September 2009	Donut	4
2.0 - 2.1	26 Oktober 2009	Éclair	5/6/7
2.2	20 Mei 2010	Froyo	8
2.3 - 2.3.7	6 Desember 2010	Gingerbread	9/10
3.0 - 3.2	22 Februari 2011	Honeycomb	11/12/13
4.0 - 4.04	19 Oktober 2011	Ice Cream Sandwich	14/15
4.1 - 4.3	9 Juli 2012	Jelly Bean	16/17/18
4.4	31 Oktober 2013	Kitkat	19
5.0	15 Oktober 2015	Lolipop	21
6.0	19 Agustus 2015	Marshmallow	23
7.0	22 Agustus 2016	Noughat	24

2.2.1 Android SDK (Software Development Kit)

Android SDK adalah *tools* API (*Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *Middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK *(Software Development Kit)* sebagai alat bantu dan API untuk mulai mengembangkan aplikasi

pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai platform apliasi-netral, Android memberi Anda kesempatan untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur-fitur Android yang paling penting adalah :

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile*
3. *Integrated browser* berdasarkan *engine open source Webkit*.
4. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi *hardware*).
5. SQLite untuk penyimpanan data.
6. *Media Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM *Telephony* (tergantung *hardware*).
7. *Bluetooth*, EDGE, 3G dan WiFi (tergantung *hardware*).
8. Kamera, GPS, kompas, dan *accelerometer* (tergantung *hardware*).
9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator, tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE Android Studio.[2]

2.2.2 ADT (Android Development Tools)

Android Development Tools (ADT) adalah plugin yang didesain untuk IDE Android Studio yang memberikan kita kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Android Studio. Dengan menggunakan ADT untuk Android Studio akan memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan *running* aplikasi menggunakan Android SDK melalui Android Studio. Dengan ADT juga kita dapat melakukan pembuatan *package* Android (.apk) yang digunakan untuk distribusi aplikasi Android yang kita rancang.

Mengembangkan aplikasi Android dengan menggunakan ADT di Android Studio sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi Android. Berikut adalah versi ADT yang sudah dirilis:

1. ADT 12.0.0 (July 2011)
2. ADT 11.0.0 (June 2011)

3. ADT 10.0.1 (March 2011)
4. ADT 10.0.0 (February 2011)
5. ADT 9.0.0 (January 2011)
6. ADT 8.0.1 (December 2010)
7. ADT 8.0.0 (December 2010)
8. ADT 0.9.9 (September 2010)
9. ADT 0.9.8 (September 2010)
10. ADT 0.9.7 (May 2010)
11. ADT 0.9.6 (March 2010)
12. ADT 0.9.5 (December 2009)
13. ADT 0.9.4 (October 2009)

Semakin tinggi *platform* Android yang kita gunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya munculnya *platform* baru diikuti oleh munculnya versi ADT yang terbaru.[2]

2.3 Pemrograman Berorientasi Objek *(Object Oriented Programming)*

Pemogram berorientasi objek adalah salah satu bentuk pemograman komputer yang bertitik tolak pada konsep objek. Pendekatan pemograman pada konsep objek masih menggunakan pendekatan terstruktur yaitu dimana bahasa program dibuat berdasarkan urutan login yang terstruktur, tetapi program yang dibuat bertitik tolak pada keberadaan objek yang ada. Pemograman pada objek ditujukan untuk mengatur sifat-sifat (properties) sebuah objek.

Secara spesifik, pengertian berorientasi objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemrograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah.[3] Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, *polymorphism* (polimorfisme), dan *inheritance* (pewarisan)

2.4 Unified Modeling Language (UML)

UML adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banuak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemanduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML[4]. Standar UML dikelola oleh OMG (*Object Management Group*).

UML bukanlah bahasa pemrograman visual, tapi bahasa pemodelan visual, bukanlah spesifikasi kakas, tapi spesifikasi bahasa pemodelan, bukanlah proses, tapi yang memungkinkan proses-proses. Terdapat perbedaan antara metode dan bahasa pemodelan. Metode adalah cara eksplisit yang menstrukturkan berfikir dan aksi seseorang. Metode memberitahu ke pemakai mengenai apa yang dilakukan, bagaimana melakukan, kapan melakukan dan kenapa dilakukan (maksud aktivitas spesifik). Metode-metode menghasilkan model-model dan model-model ini digunakan untuk mendeskripsikan sesuatu dan mengkomunikasikan hasil-hasil dari penggunaan metode. Perbedaan utama antara metode dan bahasa pemodelan adalah bahasa pemodelan tidak menyatakan proses atau intruksi untuk melakukan apa, bagaimana melakukan, kapan melakukan dan kenapa dilakukan. Model diekspresikan dalam bahasa pemodelan. Bahasa pemodelan berisi notasi, yaitu simbol-simbol yang digunakan di model dan aturan-aturan yang menuntun bagaimana menggunakannya. Aturan-aturan ini termasuk sintaks, semantics dan pragmatik.

Tujuan utama perancangan UML adalah :

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.

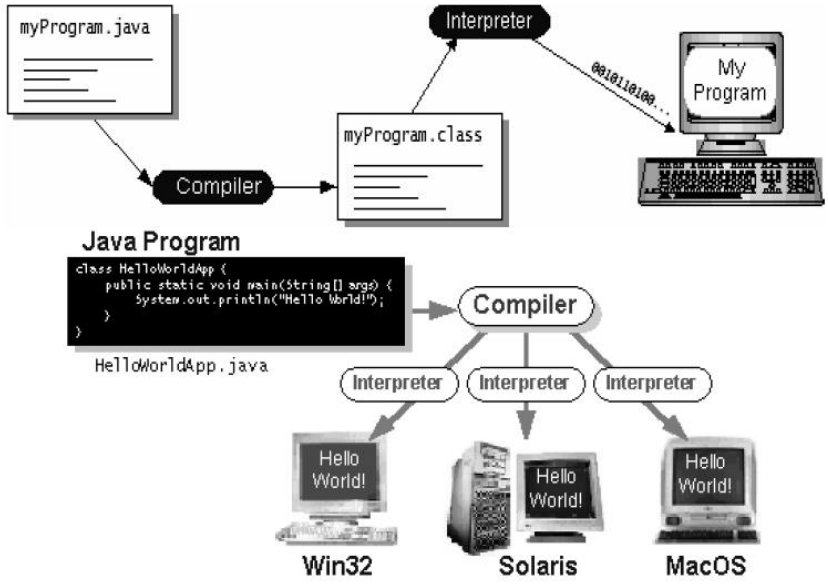
5. Mendorong pertumbuhan pasar kakas berorientasi objek.
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, framework dan patern.

Konsep-konsep yang diterapkan di UML adalah satu model berisi informasi mengenai sistem (atau domain), model-model berisi elemen-elemen model seperti kelas-kelas, simpul-simpul, paket-paket, dan sebagainya. Satu diagram menunjukkan satu pandangan tertentu dari model.

2.5 Bahasa Pemrograman Java

Saat ini Java merupakan salah satu bahasa pemrograman paling populer untuk segala hal mulai dari permainan pemrograman untuk menciptakan aplikasi *mission-critical.* Tujuannya adalah untuk mempermudah dan menjadi platform independen alternative menggunakan C++. Java adalah program yang sangat populer di lingkungan pendidikan saat ini, hampir di seluruh kampus memiliki satuan ajar pemrograman ini. *Java Virtual Machine* (JVM) atau *Java Runtime Environment* (JRE) merupakan fitur Java yang membuatnya dapat berjalan pada semua *platform* sistem operasi[5]. Apabila membuat Java pada Windows, dapat pula dijalankan pada sistem operasi Linux, Mac OS, dan lainnya.

Hal yang paling mendasar dan harus dikuasai oleh Programer Java adalah memasang library (kelas dalam bentuk biner). Karena java yang dirancang secara orientasi objek, tentunya program yang terdiri atas kesatuan antar objek yang membentuk aplikasi. Sehingga sebuah objek tertentu bisa digunakan dalam beberapa aplikasi yang berbeda. Objek ini biasanya bersifat umum penggunaannya, misalkan objek kalender, objek *driver* basis data, objek *report* dan lain-lain.



Gambar 2.1 Cara Kerja Pemrograman Java

2.6 Basis Data

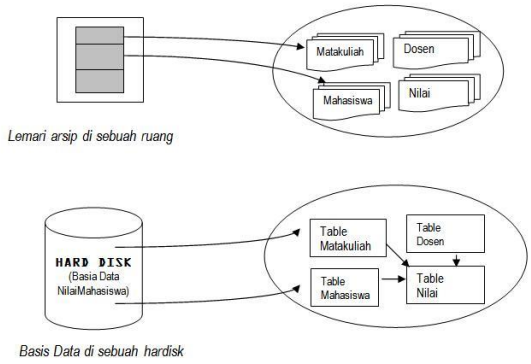
Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah refresentasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi, berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data dan lemari arsip memiliki prinsip dan tujuan yang hampir sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip menggunakan lemari sebagai media penyimpanan, maka basis data menggunakan media

penyimpanan elektronis seperti disk (disket dan *harddisk*). Tetapi, satu hal yang harus diperhatikan, bahwa basis data bukan hanya sekedar penyimpanan data secara elektronis. Artinya, tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data[6].



Gambar 2.2 Lemari Arsip dan Basis Data

2.7 SQLite

SQLite adalah salah satu *software* yang *embedded* yang sangat populer, kombinasi SQL *interface* dan penggunaan memori yang sangat sedikit dengan kecepatan yang sangat cepat. SQLite di Android termasuk dalam Android *Runtime*, sehingga setiap versi dari Android dapat membuat database dengan SQLite[7]. Dalam sistem Android memiliki beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut:

1. *Shared Prefences* yaitu menyimpan data beberapa nilai (*value*) dalam bentuk *groups key* yang dikenal dengan *prefences*.
2. *Files* yaitu menyimpan data dalam bentuk file, dapat berupa menulis ke file atau membaca dari file.
3. *SQLite Databases* yaitu menyimpan data dalam bentuk *Databases.*
4. *Content Providers* yaitu menyimpan data dalam bentuk *content providers service.*