

BAB II

DASAR TEORI

Bab ini membahas dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai Rancang Bangun Sistem Administrasi toko UD Sumber Tani berbasis ERP.

2.1 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan [IAN – 11:23].

Secara fungsional perangkat lunak yang baik adalah perangkat lunak yang bisa menjalankan semua perintah yang dilakukan oleh pengguna namun, dalam menjaga persaingan kesuksesan suatu perangkat lunak perlu untuk diperhatikan aspek – aspek nonfungsional perangkat lunak. Berikut adalah beberapa aspek nonfungsional yang perlu diperhatikan menurut Sommerville [IAN - 11] :

- a. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (*maintainability*)
- b. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability and robust*)
- c. Efisien dari segi sumber daya dan penggunaan
- d. Kemudahan perangkat lunak untuk digunakan sesuai keperluan dari pengguna (*usability*)

Kesimpulan dari beberapa aspek di atas, perangkat lunak yang baik adalah perangkat lunak yang mampu memenuhi kebutuhan dari *customer* atau pemakai dari perangkat lunak (*user*) baik secara fungsional dan nonfungsional sehingga perangkat lunak harus berorientasi pada pelanggan atau pemakai perangkat lunak, bukan berorientasi pada pembuat atau pengembang perangkat lunak.

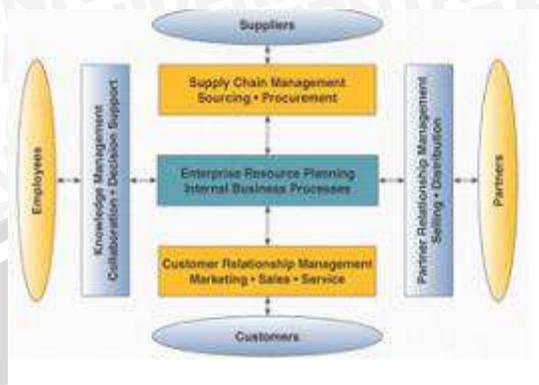
2.2 ERP

ERP merupakan sebuah kerangka kerja transaksi berskala perusahaan yang menghubungkan proses pemesanan barang, manajemen inventarisasi dan kontrol,

perencanaan distribusi dan produksi, dan keuangan. ERP bekerja sebagai kekuatan lintas fungsional perusahaan yang mengintegrasikan dan mengotomatisasi berbagai proses



bisnis internal dan sistem informasi termasuk manufaktur, logistik, distribusi, akuntansi, keuangan, dan sumber daya manusia dari sebuah perusahaan [ENT-02:35-37].



Gambar 2.1 Konsep ERP

Sumber : [RES-12]

Berdasarkan gambar 2.1 terlihat bahwa sistem ERP mengintegrasikan informasi dan proses-proses yang berbasis informasi pada sebuah bagian atau antar bagian dalam suatu organisasi atau perusahaan. Sistem ERP terdiri atas beberapa sub sistem / modul yaitu sistem finansial, sistem distribusi, sistem manufaktur, sistem *inventory*, dan sistem sumber daya manusia. Masing-masing sub sistem terhubung dengan sebuah basis data terpusat yang menyimpan berbagai informasi yang dibutuhkan oleh masing-masing sub sistem. Sub sistem mewakili sebuah bagian fungsionalitas dari sebuah organisasi perusahaan.

Sistem ERP memiliki beberapa karakteristik sebagai berikut [ETE - 13]:

- a. Sistem ERP merupakan paket perangkat lunak yang didesain pada lingkungan *client-server* baik tradisional (berbasis *desktop*) maupun berbasis *web*.
- b. Sistem ERP mengintegrasikan mayoritas proses bisnis yang ada.
- c. Sistem ERP memproses seluruh transaksi organisasi perusahaan.
- d. Sistem ERP menggunakan basis data skala *enterprise* untuk penyimpanan data.
- a) Sistem ERP memungkinkan pengguna mengakses data secara *real time*.

Sistem ERP dalam skripsi ini adalah sebuah sistem informasi pertokoan yang dirancang untuk mengkoordinasikan semua sumber daya dalam hal ini stok obat dan supplier serta informasi yang diperlukan untuk proses bisnis dalam skala kecil. Sistem ERP merupakan perangkat lunak yang mengintegrasikan semua aktor pertokoan ke

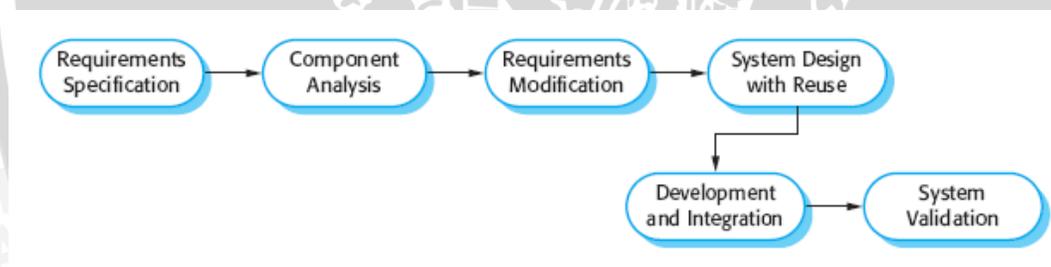
dalam satu sistem komputer yang dapat melayani semua kebutuhan pertokoan, baik dari bagian penjualan dan bagian administrator.

Alasan lain penggunaan ERP adalah integrasi. Integrasi yang dimaksud adalah menggabungkan berbagai kebutuhan pada satu perangkat lunak dalam satu *logical database*, sehingga memudahkan semua departemen berbagi informasi dan berkomunikasi.

2.3 Reuse-Oriented Model

Model proses rekayasa perangkat lunak dibentuk dengan menyesuaikan sifat dari aplikasi yang akan dibuat. Salah satu dari model proses yang digunakan adalah *Reuse-oriented*. *Reuse-oriented* merupakan rekayasa perangkat lunak yang menggunakan pendekatan berbasis tersedianya jumlah komponen *reusable* yang signifikan. Proses pengembangan sistem berfokus pada integrasi komponen-komponen ini ke dalam sistem daripada mengembangkan sistem dari dasar [IAN-1:20].

Model proses secara umum dari pengembangan berbasis *reuse* dapat dilihat pada gambar berikut :



Gambar 2.2 *Reuse-Oriented Model*

Sumber : [IAN – 11:35]

Secara sepintas model di atas sama dengan model perancangan perangkat lunak biasa, namun sebenarnya terdapat perbedaan pada pertengahan proses, yaitu dengan adanya proses *reuse-oriented*. Menurut *Sommerville* [IAN - 11] proses ini antara lain:

1. Analisis komponen. Dari spesifikasi kebutuhan yang telah ditentukan, dicari komponen yang dapat mengimplementasi spesifikasi tersebut. Biasanya, tidak ada komponen yang benar-benar mampu memenuhi spesifikasi dan mungkin hanya beberapa fungsi yang digunakan dari komponen tersebut.

2. Modifikasi kebutuhan. Pada proses ini, kebutuhan dianalisis menggunakan informasi tentang komponen yang akan digunakan. Kemudian, kebutuhan ini dimodifikasi untuk merefleksikan komponen yang tersedia. Jika terdapat modifikasi yang tidak mungkin untuk diterapkan, proses analisis komponen dapat dilakukan kembali untuk mencari solusi alternatif.
3. Desain sistem dengan *reuse*. Pada proses ini, *framework* dari sistem atau *framework* yang telah ada digunakan kembali (*reused*).
4. Pengembangan dan integrasi.

Sommerville [IAN - 11] juga menyebutkan tiga tipe komponen perangkat lunak yang dapat digunakan dalam proses *reuse-oriented*, yaitu :

1. *Web service* yang dikembangkan berdasarkan *service standard* dan tersedia untuk digunakan secara *remote*.
2. Koleksi *object* yang dikembangkan sebagai *package* untuk diintegrasikan dengan sebuah *framework* komponen seperti *.NET* dan *J2EE*.
3. Sistem perangkat lunak yang berdiri sendiri yang dapat dikonfigurasi untuk digunakan dalam lingkungan tertentu.

Rekayasa perangkat lunak yang *reuse-oriented* memiliki keuntungan dalam mengurangi jumlah perangkat lunak yang dikembangkan dan mengurangi biaya dan resiko. *Reuseable* yang digunakan nantinya adalah kerangka kerja AhadPOS yang dikembangkan untuk merancang aplikasi agar dapat digunakan kembali pada tahap pengembangan. Secara detail AhadPOS dapat dijelaskan pada sub bab 2.4.

2.3 Unified Modelling Language

Unified Modelling Language (UML) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung yang memudahkan untuk dipahami [IBM-01:6-9]. *UML* ada untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasi sistem perangkat lunak yang dikembangkan. *UML* hanya berfungsi untuk melakukan pemodelan, sehingga penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek [ROS-11].

Kategori *UML* dibedakan menjadi 3 kategori besar yakni:

- a. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- b. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- c. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsystem pada suatu sistem.

2.3.1 Usecase Diagram

Usecase merupakan pemodelan untuk perilaku (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [ROS-11].

Syarat penamaan pada *usecase* adalah nama didefinisikan sesederhana mungkin dan mudah dipahami. Dua hal utama *usecase* adalah pendefinisian apa yang disebut aktor dan *usecase*.

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Usecase* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor.

2.3.2 Activity Diagrams

Activity diagrams menggambarkan aliran aktivitas dalam sistem yang sedang dirancang. Hal – hal yang perlu diperhatikan dalam perancangan *activity diagrams* adalah bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagrams* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi alasannya karena *activity diagrams* merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Gambaran proses bisnis dan urutan aktivitas dalam sebuah proses yang digunakan pada *business modeling* dapat memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *Flowchart* atau *Data Flow Diagram* pada perancangan terstruktur sehingga sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan [MEI -13].

2.4 AhadPOS

AhadPOS adalah *frameworkPOS / Point Of Sales (Framework kasir / jual-beli)* untuk UKM, dengan lisensi *Open Source (GPL v2)* yang mulai berkembang di Indonesia sejak tahun 2010. Framework ini bersifat open bagi semua kalangan sehingga siapa saja diperbolehkan untuk menggunakan dan memodifikasi sistem secara maksimal [AHA-13]. Secara garis besar AhadPOS memiliki beberapa modul yang dikembangkan yakni modul *inventory* yang mencakup semua data barang yang tersedia, modul *sales* yang mencakup semua data penjualan dan pembelian barang, dan modul *control* yang mengatur semua data pengguna sistem.

AhadPOS di desain untuk dapat [AHA-13]:

- Melakukan banyak transaksi jual-beli dengan akurat, mudah, dan cepat : memungkinkan UKM untuk berkembang.
- Memungkinkan pelacakan stok / *inventory* yang akurat secara real-time : meminimalisir kemungkinan kebocoran, kecurangan, pencurian, dll
- Pencatatan transaksi keuangan yang akurat : salah satu masalah di UKM adalah pembukuan. Dengan AhadPOS, maka ini langsung tercatat di *database*-nya secara otomatis, sehingga bisa terlacak secara akurat.

Dari kelebihan – kelebihan itulah maka framework ini dipilih untuk digunakan pada sistem UD. Sumber Tani Batu dalam perancangan dan pengembangan sistem.

2.5 MySQL - Xampp

MySQL- Xampp adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS* yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh

dunia. Aplikasi ini menggunakan *databaseMySQL* karena *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* [WKI-13].

Menurut [WKI – 13]*MySQL-Xampp* memiliki beberapa keistimewaan, antara lain:

- a. Portabilitas. *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
- b. *Open Source*. *MySQL-Xampp* di distribusikan secara *open source*, dibawah lisensi *GPL* sehingga dapat digunakan secara cuma-cuma.
- c. '*Multiuser*'. *MySQL-Xampp* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- d. '*Performance tuning*'. *MySQL-Xampp* memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.
- e. Jenis Kolom. *MySQL* memiliki tipe kolom yang sangat kompleks, seperti *signed* atau *unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
- f. Perintah dan Fungsi. *MySQL-Xampp* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
- g. Keamanan. *MySQL-Xampp* memiliki beberapa lapisan keamanan seperti *level subnetmask*, nama *host*, dan izin akses pengguna dengan sistem perizinan yang mendetail serta sandi terenkripsi.
- h. Skalabilitas dan Pembatasan. *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas *indeks* yang dapat di tampung mencapai 32 *indeks* pada tiap tabelnya.
- i. Konektivitas. *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol *TCP/IP, Unix soket (UNIX)*, atau *Named Pipes(NT)*.
- j. Antar Muka. *MySQL-Xampp* memiliki antarmuka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *Application Programming Antarmuka*.

- k. Klien dan Peralatan. *MySQL-Xampp* dilengkapi dengan berbagai peralatan (*tools*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
1. Struktur tabel. *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *Postgre SQL* ataupun *Oracle* [WKI-13].

Alasan skripsi ini menggunakan *MySQL- Xampp* karena produk ini *open source* dan sudah sangat populer dalam aplikasi *web* seperti MediaWiki (perangkat lunak yang dipakai Wikipedia dan proyek-proyek sejenis) dan *PHP-Nuke* dan berfungsi sebagai komponen basis data dalam *LAMP*. Popularitas sebagai aplikasi *web* dikarenakan kedekatannya dengan popularitas *PHP*, sehingga seringkali disebut sebagai *Dynamic Duo*[WKI-13].

2.6 HTML/PHP

HyperText Markup Language(HTML) adalah sebuah bahasa yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web internet* dan pemformatan *hiperteks* sederhana yang ditulis dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format *ASCII* normal sehingga menjadi halaman *web* dengan perintah-perintah *HTML* [WIK-13].

PHP merupakan singkatan rekursif (akronim berulang) dari *PHP Hypertext Preprocessor*. *PHP* adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini atau dalam kata lain bisa diartikan sebuah bahasa pemrograman *web* yang bekerja di sisi *server (server side scripting)* yang dapat melakukan konektivitas pada *database* dimana hal itu tidak dapat dilakukan hanya dengan menggunakan sintaks-sintaks *HTML* biasa. *PHP* banyak dipakai untuk memprogram situs *web* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain [NUR-13].

2.7 Thin Client Server Computing

Jaringan *Thin Client* adalah suatu lingkungan jaringan, yang mana *client* berfungsi sebagai terminal yang mengakses data dan aplikasi dari komputer *server*.

Secara terpusat pengolahan data dilakukan oleh *server* dan *client* hanya memproses *input* dari *keyboard*, *mouse*, dan keluaran berupa tampilan atau gambar (*display*), hal ini karena proses seutuhnya dilakukan oleh *server* [THI-13]. *Server* utama menyediakan aplikasi dan sumber daya lainnya untuk sejumlah besar Terminal. Terminal (*client*) hanya cukup mengoperasikan *mouse*, *keyboard* dan *monitor*, *client* dapat menjalankan berbagai aplikasi yang terpasang pada *server*. *Thin Client Server Computing* (TCSC) merupakan suatu konsep jaringan komputer yang menekankan proses komputasi pada sisi *Client* yang berkinerja seminimal mungkin.

Pada skripsi ini metode TCSC ini digunakan dalam rencana implementasi aplikasi yang dibuat pada toko UD. Sumber Tani Batu.



