BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Konsep dasar program terdapat beberapa definisi yang berhubungan dengan pengertian aplikasi, pengertian program, database (basis data), dan pengertian piutang usaha.

2.1.1. Pengertian Aplikasi

Aplikasi menurut Hendrayudi (2008:143) “adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus).” Sedangkan menurut Maryono dan Istiana (2008:8) “Aplikasi adalah program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu”

Dari pengertian Aplikasi diatas dapat disimpulkan bahwa Aplikasi adalah kumpulan dari perintah program yang dibuat dan dikembangkan untuk memenuhi kebutuhan pengguna (*user*) agar mempermudah dalam menggunakan suatu program.

2.1.2. Pengertian Program

Menurut Irwansyah dan Moniaga (2014:13) mengemukakan: “*Software* atau biasa disebut program adalah deretan dari perintah yang berhubungan dan teratur untuk mencapai tujuan tertentu yang memberitahu komputer apa yang harus dilakukan dan bagaimana melakukannya.”

Menurut Kadir (2015:3) “program tak lain adalah kumpulan instruksi yang ditujukan untuk komputer supaya peralatan tersebut dapat melakukan tindakan-tindakan yang dikehendaki oleh pemakai program (*user*). Adapun orang yang membuat program dinamakan pemogram (*programmer*).”

Dari pengertian program diatas dapat disimpulkan bahwa program adalah deretan dari perintah yang berhubungan dan teratur untuk mencapai tujuan tertentu yang memberitahu komputer apa yang harus dilakukan dan bagaimana melakukannya sehingga dapat menampilkan kumpulan instruksi yang ditujukan untuk komputer supaya peralatan tersebut dapat melakukan tindakan-tindakan yang dikehendaki oleh pemakai program (*user*).

2.1.3. Pengertian Database

Menurut Anhar (2010:45) mengemukakan *Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari field atau kolom.”

Menurut Sukamto dan Shalahuddin (2015:44) DBMS (*Database managment system*) atau dalam bahasa Indonesia sering disebut Sistem manajemen basis data adala suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data.”

Dari pengertian *Database* diatas dapat disimpulkan bahwa *database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari field atau kolom dan suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data.

2.1.4. Pengertian Piutang Usaha

Menurut Waluyo (2008:58) “Piutang Usaha (*account receivable*) meliputi piutang yang timbul karena adanya penjualan produk atau penyerahan jasa dalam rangka kegiatan usaha normal perusahaan.” Sedangkan Menurut Supriyati (2016:16) “Piutang Dagang/Piutang Usaha (*account receivable*) adalah piutang yang berasal dari penjualan barang dagangan kepada pihak pelanggan yang dilakukan secara kredit. Sedang, bila transaksi tersebut berkaitan dengan penyerahan jasa dengan pembayaran kemudian dikelompokkan sebagai piutang usaha. Piutang dagang maupun piutang usaha dikelompokkan sebagai unsur aset lancar dalam laporan posisi keuangan. Dari pengertian Piutang Usaha (*account receivable*) dapat disimpulkan bahwa Piutang Usaha (*account receivable*) meliputi piutang yang timbul karena adanya penjualan produk atau penyerahan jasa dalam rangka kegiatan usaha normal dan berasal dari penjualan barang dagangan kepada pihak pelanggan yang dilakukan secara kredit dan transaksi tersebut berkaitan dengan penyerahan jasa.

2.2. Peralatan Pendukung (*Tools Program*)

Peralatan Pendukung (*Tools Program*) yang penulis gunakan dalam pembuatan meliputi OOP (*Object Oriented Programming*), Java, UML (*Unified Modeling Language*), *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Deployment Diagram*, *Sequence Diagram*, Netbeans, Xampp, Mysql, dan peralatan lainnya yang membantu penulis dalam pembuatan Tugas Akhir (TA).

2.2.1. *Object Oriented Programming (OOP)*

Object Oriented Programming (OOP) menurut Andi (2010:88) “adalah suatu cara mengorganisasi program dengan memodelkan objek-objek dunia nyata (seperti benda, sifat, sistem, dan lainnya) ke dalam sebuah bahasa pemrograman. Dalam bahasa pemrograman pemodelan OOP ini diaplikasikan dengan sebuah kelas, method, dan properti/variabel.” Sedangkan menurut Wahana Komputer (2010:59) “Pemrograman berorientasi objek atau *Object oriented programming (OOP)* adalah peningkatan dari pendekatan pemrograman prosedural yang telah bertahun-tahun dianut oleh kalangan pengembang *software*. Terutama yang masih menggunakan *software-software developer* lama seperti C, C++, Pascal, Basic, ADA, dan COBOL.”

Dari pengertian *Object Oriented Programming (OOP)* dapat disimpulkan bahwa *Object Oriented Programming (OOP)* adalah suatu cara mengorganisasi program dengan memodelkan objek-objek dunia nyata yang merupakan peningkatan dari pendekatan pemrograman prosedural yang telah bertahun-tahun dianut oleh kalangan pengembang *software*.

2.2.2. JAVA

Menurut Sukamto dan Shalahuddin (2015:103) “Java merupakan bahasa pemrograman murni karena semua kode programnya dibungkus dalam kelas.”

Menurut Nofriadi (2015:1) “Bahasa Pemrograman Java merupakan salah satu bahasa pemrograman yang dapat dijalankan di berbagai sistem operasi termasuk telepon genggam. Bahasa pemrograman ini merupakan pengembangan dari bahasa pemrograman C++ karena banyak mengadopsi sintak C dan C++.”

Dari pengertian Java diatas dapat disimpulkan bahwa Java merupakan bahasa pemrograman murni dan merupakan salah satu bahasa pemrograman yang dapat dijalankan di berbagai sistem operasi termasuk telepon genggam, dan juga merupakan pengembangan dari bahasa pemrograman C++ karena banyak mengadopsi sintak C dan C++”.

2.2.3. *Unified Modeling Language (UML)*

Menurut Nugroho (2010:6) “*Unified Modeling Language (UML)* adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’.”

”UML Merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi pengguna UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. Seperti yang kita ketahui bahwa banyak hal di dunia sistem informasi yang tidak dapat dibakukan, semua tergantung kebutuhan, lingkungan dan konsteksnya. Begitu juga dengan perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi belum tentu pandangan yang berbeda dalam penggunaan UML adalah suatu salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin divisualkan. Secara analogi jika dengan bahasa yang kita gunakan sehari-hari, belum tentu penyampaian bahasa dengan puisi adalah hal yang salah. Sistem informasi bukanlah ilmu pasti, maka jika ada banyak perbedaan dan

interpretasi didalam bidang sistem informasi merupakan hal yang sangat wajar.” (Sukanto dan Shalahuddin, 2015:137).

Dari pengertian *Unified Modeling Language (UML)* dapat disimpulkan *Unified Modeling Language (UML)* adalah bahasa pemodelan untuk sistem atau perangkat lunak dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung yang berparadigma berorientasi objek.

Beberapa diagram yang ada pada UML, diantaranya:

1. *Use Case Diagram*

Menurut Nugroho (2009:10) “*Use Case Diagram* merupakan gambaran menyeluruh dan pada umumnya sangatlah tidak terperinci oleh karena itu kita harus memperinci lagi perilaku sistem untuk masing-masing *use case* yang ada.”

Menurut Sukanto dan M. Shalahuddin (2015:155) “*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 berhak menggunakan fungsi-fungsi itu.”

Dari pengertian *Use Case* diatas dapat disimpulkan bahwa *Use Case* adalah Gambaran menyeluru dan pada umumnya sangat tidak terperinci oleh karena itu kita harus memperinci lagi perilaku sistem untuk masing-masing *use case* yang ada pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.

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*.

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

2. *Activity Diagram*

Menurut Nugroho (2010:62) “Diagram aktivitas (*activity diagram*) sesungguhnya merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan.”

Menurut Sukamto dan Shalahuddin (2015:161) “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.”

Dari pengertian Diagram Aktivitas (*activity diagram*) diatas dapat disimpulkan bahwa Diagram Aktivitas (*activity diagram*) sesungguhnya merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan sehingga menggambarkan *workflow* (aliran kerja)

atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak yang menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokkan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- d. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

3. *Class Diagram*

Menurut McLeod, dan Schell (2008:176) “Diagram Kelas (*Class Diagram*) terdiri atas kelas-kelas yang memiliki nama, *field-field* didalam kelas, dan tindakan-tindakan (kadang disebut sebagai metode) yang dilakukan atas kelas.”

Menurut Sukamto dan Shalahuddin (2015:141) mengemukakan “Kelas Diagram 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.

- a. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.”

Dari pengertian Kelas Diagram atau *Class Diagram* diatas dapat disimpulkan bahwa Kelas Diagram atau *Class Diagram* merupakan terdiri atas kelas-kelas yang memiliki nama, *field-field* didalam kelas, dan tindakan-tindakan yang dapat menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang memiliki atribut dan metode.

4. *Sequence Diagram*

Menurut Sulistyorini (2009:27) mengemukakan “*Diagram sequence* menjelaskan interaksi objek yang disusun dalam suatu urutan objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use case*. *Sequence diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*. Diagram *sequen* sebaiknya digunakan diawal tahap desain atau analisis karena kesederhaannya dan mudah untuk dimengerti.”

Menurut Sukamto dan Shalahuddin (2015:185) “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram *sekuen* juga dibutuhkan untuk melihat sekenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup

pada diagram sekuen sehingga semakin banyak *use case* yang telah didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.”

Dari pengertian *Diagram sequence* dapat disimpulkan bahwa “*Diagram sequence* menjelaskan interaksi objek yang disusun dalam suatu urutan objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use case* dan menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

2.2.4. Netbeans

Menurut Nofriadi (2015:4) “Netbeans merupakan sebuah aplikasi *Integrated Development Environment* (IDE) yang berbasiskan Java dari *Sun Microsystems* yang berjalan diatas *swing* dan banyak digunakan sekarang sebagai editor untuk berbagai bahasa pemrograman.” Sedangkan menurut Andi (2010:2) “Netbeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu Netbeans IDE dan Netbeans Platform. Netebeans IDE merupakan produk yang digunakan untuk melakukan pemograman baik menulis kode, mengompilasi, mencari kesalahan, dan mendistribusikan program. sedangkan Netbeans Platform adalah sebuah modul yang merupakan kerangka awal/fondasi dalam membangun aplikasi *dekstop* yang besar.”

Dari pengertian Netbeans dapat disimpulkan bahwa Netbeans adalah sebuah aplikasi Java dari *Sun Microsystems* yang berjalan diatas *swing* dan merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, mengompilasi, mencari kesalahan, dan mendistribusikan program, termasuk program aplikasi dekstop.

2.2.5. Xampp

Menurut Aditya (2011:16) “Xampp adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan komplikasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, My SQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama Xampp merupakan singkatan 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 digunakan yang dapat melayani tampilan halaman web yang dinamis.” Sedangkan menurut Kurniawan (2008:1) “Xampp adalah sebuah aplikasi instan dan lengkap dikarenakan segala yang anda butuhkan untuk membuat sebuah situs web dengan Word-Press terdapat dalam aplikasi ini.”

Dari pengertian Xampp dapat disimpulkan bahwa “Xampp adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan komplikasi dari beberapa program dan merupakan sebuah aplikasi instan dan lengkap dikarenakan segala yang anda butuhkan untuk membuat sebuah situs web dengan Word-Press terdapat dalam aplikasi Xampp ini.

2.2.6. Mysql (*My Structure Query Language*)

Menurut Aditya (2011:61) “MySQL (*My Structure Query Language*) 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 diseluruh dunia. MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public Licence*).”

Menurut Anhar (2010:21) “MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (Database Management System) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL, dan lain-lain. MySQL merupakan DBMS yang multithread, multi user yang bersifat gratis dibawah lisensi GNU *General Public Licence* (GPL).”

Dari pengertian MySQL (*My Structure Query Language*) dapat disimpulkan bahwa MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL sebuah implementasi dari sistem manajemen basis data relasional (RDMS), dan DBMS yang multithread, multi user yang bersifat gratis dibawah lisensi GNU *General Public Licence* (GPL).

Beberapa kelebihan MySQL, antara lain:

- a. MySQL dapat berjalan dengan stabil pada berbagai sistem operasi, seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, dan masih banyak lagi
- b. Bersifat Open source, MySQL didistribusikan secara open source (gratis), dibawah lisensi GNU *General Public Licence* (GPL).

- c. Bersifat Multiuser, MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengaami masalah.
- d. MySQL memiliki kecepatan yang baik dalam menangani query (perintah SQL). Dengan kata lain, dapat memproses lebih banyak SQL per satuan waktu.
- e. Dari segi *security* atau keamanan data, MySQL memiliki beberapa lapisan security, seperti level subnet mask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password yang terenkripsi.
- f. Selain MySQL bersifat fleksibel dengan berbagai pemograman, MySQL juga memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemograman dengan menggunakan fungsi API (*Application Programming Interface*).