

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Penjelasan mengenai definisi-definisi serta uraian singkat mengenai konsep dasar dari program yang berhubungan dengan tugas akhir.

2.1.1. Pengertian Program

Menurut Kadir (2010:2), program adalah “kumpulan perintah yang ditunjukkan kepada komputer agar komputer dapat melakukan tindakan sesuai yang dikehendaki oleh pembuat perintah.”

Sedangkan pengertian program menurut Kurniawan (2011:65) berpendapat bahwa program adalah “kumpulan instruksi yang disusun dengan urutan nalar yang tepat untuk menyelesaikan suatu persoalan.”

Dari pengertian diatas dapat disimpulkan bahwa program adalah kumpulan instruksi atau perintah yang ada didalam komputer dibuat oleh pembuat perintah untuk menyelesaikan suatu persoalan.

2.1.2. Pengertian Aktiva

Menurut Himayati (2007:3) mengemukakan bahwa aktiva adalah “sumber daya yang dimiliki oleh entitas bisnis atau usaha.” Sumber daya ini dapat berbentuk fisik ataupun hak yang mempunyai nilai ekonomis. Sedangkan menurut Shatu (2016:73) aktiva (*assets*) adalah “harta kekayaan yang dimiliki oleh suatu perusahaan.”

Berdasarkan pengertian aktiva diatas, dapat disimpulkan bahwa aktiva adalah harta kekayaan yang dimiliki oleh perusahaan dapat berbentuk fisik ataupun hak yang mempunyai nilai ekonomis.

Aktiva dapat dibedakan atas aktiva lancar dan aktiva tetap menurut Ahman dan Indriani (2007:156).

1. Aktiva Lancar (*Current Assets*)

Aktiva lancar, yaitu aktiva yang dapat di ubah menjadi uang, dijual, atau dipakai dalam jangka waktu satu periode akuntansi.

2. Aktiva Tetap (*Fixed Assets*)

Aktiva tetap, yaitu aktiva yang memiliki manfaat ekonomis lebih dari satu periode akuntansi (lebih dari satu tahun).

Akun aktiva tetap dapat dikelompokkan menjadi tiga, yaitu investasi jangka panjang (*long term investment*), aktiva tetap berwujud (*tangible fixed assets*), dan aktiva tetap tidak berwujud (*intangible fixed assets*).

a. Investasi Jangka Panjang

Aktiva yang berupa invenstasi dalam bentuk saham atau obligasi serta setoran simpanan wajib ke koperasi induk yang jangka waktu penanaman atau penyimpanannya lebih dari satu tahun atau untuk jangka panjang.

b. Aktiva Tetap Berwujud

Aktiva perusahaan yang wujudnya dapat dilihat. Aktiva ini memiliki masa manfaat lebih dari satu tahun atau satu periode akuntansi.

c. Aktiva Tetap Tidak Berwujud

Yaitu aktiva yang wujud fisiknya tidak dapat dilihat atau tidak tampak.

2.1.3. Pengertian *Database*

Menurut Kusriani dan Koniyo (2007:141) *database* adalah “kumpulan dari item data yang saling berhubungan satu sama lain, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer, dan harus menggunakan *software* untuk melakukan manipulasi tertentu.”

Menurut Zaki dan SmitDev Community (2009:5) *database* adalah “tempat yang digunakan untuk menyimpan data.”

Dapat disimpulkan bahwa *database* adalah tempat untuk penyimpanan data-data dan harus menggunakan aplikasi untuk memanipulasi data-data tersebut.

2.1.4. *Object Oriented Programming* (OOP)

Menurut Wardana (2008:7) berpendapat bahwa, “Pemrograman Berorientasi Objek (PBO) atau *Object Oriented Programming* (OOP) adalah teknik pembuatan program yang berorientasi objek.”

Sedangkan menurut Supardi (2010:320) OOP (*Object Oriented Programming*) “merupakan cara berpikir, pandangan atau paradigma baru untuk membuat program atau merancang sistem dengan memperhatikan objek, ciri objek dan perilakunya.”

Dari pendapat diatas dapat disimpulkan bahwa *Object Oriented Programming* (OOP) adalah pembuatan program yang berorientasi objek.

2.1.4.1. Konsep Dasar Berorientasi Objek

Menurut Sukanto dan Shalahuddin (2014:103), “Pendekatan berorientasi objek merupakan salah satu tehnik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem

lainnya)”. Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek. (Sukamto dan Shalahuddin, 2014:104) :

1. Kelas (*class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

2. Objek (*object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi tempat, kejadian, struktur, status atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasim, dan dihancurkan.

3. Metode (*method*)

Operasi atau metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk

memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

4. Atribut (*attribute*)

Adalah variabel kelas yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek.

5. Abstraksi (*abstraction*)

Prinsip untuk mempresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi atau objek lain sebagai bagian dari dirinya.

8. Antarmuka (*interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain. Antarmuka atau *interface* biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya.

9. Reusability

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

11. Komunikasi Antar Objek

Komunikasi antar-objek dilakukan lewat pesan (*message*) yang dikirim dan satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

Kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13. Package

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas bernama sama disimpan dalam *package* yang berbeda.

2.2. Peralatan Pendukung (*Tools Program*)

2.2.1. Unified Modeling Language (UML)

Menurut Sukamto dan Shalahuddin (2014:133) mengemukakan bahwa, “UML (*Unified Modeling Language*) adalah salah *standar* bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek.”

Menurut Mulyani (2016:42), “*Unified Modeling Language* selanjutnya disebut *UML* adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem.”

Dari pendapat diatas dapat disimpulkan *Unified Modeling Language* (UML) adalah teknik pengembangan sistem yang menggunakan bahasa grafis untuk membuat analisis dan desain serta menggambarkan arsitektur dalam pemograman berorientasi objek.

Menurut Sukamto dan Shalahuddin (2014:141), Diagram UML terdiri dari :

1. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefisian 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.

2. *Use Case Diagram*

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 ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

3. *Activity Diagram*

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada diperangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

4. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek. Membuat diagram sekuen dilihat dari skenario yang ada pada *use case*.

2.2.2. *NetBeans IDE*

Menurut Wahana Komputer (2010:15) *NetBeans IDE* merupakan “produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-compile, mencari kesalahan dan mendistribusikan program.”

Menurut Pratama (2010:5) *NetBeans IDE* merupakan “piranti lunak terutama untuk pengembangan aplikasi berbasis *Java*.”

Dapat disimpulkan bahwa *NetBeans IDE* adalah aplikasi berbasis *Java* yang digunakan untuk membuat program.

2.2.3. My Structured Query Language (MySQL)

Huda dan Bunafit (2010:181) berpendapat bahwa, *MySQL (My Structured Query Language)* adalah “sebuah perangkat lunak sistem manajemen basis data SQL (*Structure Query Language*) atau yang dikenal dengan DBMS(*database management system*), database ini *multithread, multi-user*.”

Menurut Wahana Komputer (2010:26) *MySQL (My Structured Query Language)* adalah “salah satu *software* sistem manajemen *database* (DBMS) *Structure Query Language* (SQL) yang bersifat *open source*.”

Dari pendapat diatas dapat disimpulkan bahwa *MySQL (My Structured Query Language)* merupakan DBMS(*Database Management System*) sebuah aplikasi manajemen basis data.

2.2.4. PHPMysqlAdmin

Menurut Rahman (2013:21) *PHPMysqladmin* adalah “sebuah *software* berbasis pemograman PHP yang dipergunakan sebagai *administrator MySQL* melalui *browser* (web) yang digunakan untuk *management database*.”

Menurut Putri dalam Hikmah dkk (2015:2) *PHPMysqlAdmin* merupakan “aplikasi yang dapat digunakan untuk membuat *database*, pengguna (*user*), memodifikasi tabel, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan perintah (*command*) SQL.”

Dapat ditarik kesimpulan bahwa *PHPMysqlAdmin* merupakan aplikasi untuk *management database* berbasis *browser* (web).

2.2.5. X(Windows/Linux) Apache MySQL PHP Perl (XAMPP)

Menurut Sidik (2014:72) XAMPP (*X(Windows/Linux) Apache MySQL PHP Perl*) merupakan “paket *server* web *PHP* dan *database MySQL* yang paling populer dikalangan pengembang web dengan menggunakan *PHP* dan *MySQL* sebagai *databasenya*.”

Menurut Aryanto (2016:4) XAMPP (*X(Windows/Linux) Apache MySQL PHP Perl*) merupakan “sebuah aplikasi perangkat lunak pemrograman dan *database* yang didalamnya terdapat berbagai macam aplikasi pemrograman seperti; *Apache HTTP Server*, *MySQL database*, bahasa pemrograman *PHP* dan *Perl*.”

Jadi, kesimpulannya adalah XAMPP (*X(Windows/Linux) Apache MySQL PHP Perl*) merupakan sebuah aplikasi *server* web *PHP* dan didalamnya sudah dipaketkan terdiri dari: *Apache HTTP Server*, *MySQL database*, bahasa pemrograman *PHP* dan *Perl*.

2.2.6. Pengujian

Menurut Simarmata (2010:301) pengujian adalah “proses eksekusi suatu program untuk menemukan kesalahan.”

Menurut Sukamto dan Shalahuddin (2014:272) berpendapat bahwa pengujian adalah “satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan.”

Dari menurut ahli diatas dapat disimpulkan bahwa pengujian adalah suatu aktifitas untuk menguji program apakah terdapat kesalahan sekaligus mengevaluasi kebenaran dari program.

Pengujian perangkat lunak adalah ”sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*)” (Sukanto dan Shalahuddin, 2014:272).

Pengujian untuk validasi memiliki beberapa pendekatan menurut Sukanto dan Shalahuddin (2014:275), sebagai berikut:

a. *Black-Box Testing* (pengujian kotak hitam)

Yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

b. *White-Box Testing* (pengujian kotak putih)

Yaitu menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan.