

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

penjelasan yang berhubungan dengan definisi-definisi pada ahli tentang konsep dasar program yang berhubungan dengan penulisan tugas akhir.

2.1.1 Pengertian Program

Menurut Kurniawan (2011:15) mengemukakan Program adalah “kumpulan instruksi yang disusun dengan urutan nalar yang tepat untuk menyelesaikan suatu persoalan”.

Sedangkan menurut Djahir dan Pratita (2014:153) berpendapat bahwa Program merupakan “kumpulan dari perintah-perintah komputer yang tersusun secara sistematis”.

Berdasarkan dari penjelasan para ahli tersebut dapat disimpulkan bahwa program merupakan susunan instruksi atau perintah yang sistematis untuk menyelesaikan suatu permasalahan.

2.1.2 Pengertian Java

Menurut Supriyatno (2010:2) mengatakan *Java* merupakan “sebuah bahasa pemrograman berorientasi objek yang dapat berjalan pada *Platform* yang berbeda, baik di *Windows*, *Linux*, serta Sistem Operasi lainnya”.

adapun Menurut Sukanto dan Shalahuddin (2014:101) berpendapat bahwa *Java* merupakan “bahasa pemrograman yang paling konsisten dalam mengimplementasikan paradigma pemrograman berorientasi objek”.

Bedasarkan beberapa pendapat para ahli tentang *java*, dapat disimpulkan bahwa *Java* adalah bahasa pemrograman yang konsisten dalam mengimplementasikan setiap paradigma pemrograman berorientasi objek dan dapat dijalankan pada platform yang berbeda.

2.1.3 OOP (*Objek Oriented Programming*)

1. Konsep Dasar OOP (*Objek Oriented Programming*)

Menurut Sukamto dan Shalahuddin (2014:103) pendekatan berorientasi objek merupakan “suatu teknik 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. Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi (Sukamto dan Shalahuddin, 2014:103).

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

a. Kelas

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. Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

b. 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, dimanipulasi, dan dihancurkan.

c. Metode (*Method*)

Operasi atau metode atau method pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari event, aktivitas atau aksi keadaan, fungsi, atau kelakuan dunia nyata.

d. Atribut (*Attribute*)

Atribut dari sebuah kelas adalah *variabel* global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dimiliki secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

e. Abstraksi (*Abstraction*)

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

f. Enkapsulasi (*encapsulation*)

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

g. Pewarisan (*inheritance*)

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

h. Antarmuka (*interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Sebuah kelas dapat mengimplementasikan lebih dari satu antarmuka dimana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode program kelas itu.

i. *Reusability*

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

j. Generalisasi dan spesialisasi

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

k. Komunikasi antar objek

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

l. Polimorfisme (*polymorphism*)

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

m. *Package*

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

2. Pengertian OOP (*Objek Oriented Programming*)

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

Ngarasan dan Ardo (2012:45) berpendapat OOP yaitu “pemrograman yang menggunakan model cara berpikir manusia”.

Dilihat dari beberapa teori yang telah dipaparkan sebelumnya, dapat disimpulkan bahwa OOP (*Objek Oriented Programming*) merupakan suatu pemrograman yang menggunakan atau dirancang dengan cara berpikir manusia dengan memperhatikan objek, ciri objek dan perilakunya.

2.1.4 Pengertian DBMS (*Database Management System*)

Menurut Yanto (2016:15) berpendapat bahwa DBMS (*Database Management System*) merupakan “paket program (*software*) yang dibuat agar memudahkan dan mengefisienkan pemasukan, pengeditan, penghapusan dan pengambilan informasi terhadap database”.

sedangkan menurut Lubis (2016:25) DBMS (*Database Management System*) adalah “perangkat lunak yang menangani semua pengaksesan *database*”.

Berdasarkan dari penjelasan yang telah diuraikan sebelumnya, dapat disimpulkan bahwa DBMS (*Database Management System*) merupakan perangkat lunak yang menangani pengaksesan database dan memberikan kemudahan dan mengefisiensikan pemasukan, pengeditan, serta penghapusan data terhadap *database*.

2.1.5 Pengertian APBDES (Anggaran Pendapatan dan Belanja Desa)

penyelenggaraan urusan pemerintah dan masyarakat diperlukan adanya keseimbangan antara perkiraan pendapatan desa dengan belanja desa yang tertuang dalam peraturan desa tentang Anggaran Pendapatan Dan Belanja Desa Parit Setia Jawai Kabupaten Sambas dengan Berdasarkan dengan peraturan dari Kepala Desa Parit Setia Kecamatan Jawai Nomor 3 Tahun 2016 tentang Anggaran Pendapatan Dan Belanja Desa tahun 2016 dalam rangka pelaksanaan otonomi desa dengan melaksanakan penyelenggaraan pemerintah dan pembangunan didesa, ini merupakan pedoman bagi pemerintah desa dalam menetapkan kebijakan yang terkait dengan pemerintah desa, baik yang berasal dari pemerintah pusat, pemerintah provinsi dan pemerintah kabupaten maupun yang berasal dari pendapatan asli desa.

2.2. Peralatan Pendukung (*Tool System*)

Peralatan pendukung merupakan media yang menjadi dasar pembuatan program serta sebagai media pembantu seorang programmer dalam menyusun dan mengatur setiap alur program mulai dari proses memasukan (*input*) hingga proses keluaran (*output*) dan diimplementasikan hasilnya, kemudian memberitahukan apa yang dibutuhkan oleh programmer agar mendapatkan hasil yang diharapkan.

Fungsi dari peralatan pendukung (*tool program*) adalah memberikan penjelasan yang menjadi pendukung dalam pembuatan aplikasi oleh programmer.

2.2.1. Pengertian Netbeans

Menurut wahana komputer (2010:16) mengatakan Netbeans IDE merupakan “Produk yang digunakan untuk melakukan promrograman baik menulis kode, mong-*compile*, mencari kesalahan dan mendistribusikan program”

Menurut Gata (2012:33) berpendapat bahwa Netbeans IDE merupakan “Editor yang mengintegrasikan perangkat alat bantu GUI (*Graphic User Interface*) yang akan memudahkan user untuk membangun aplikasi dekstop”.

Dengan demikian dapat disimpulkan bahwa Netbeans IDE merupakan Program yang digunakan untuk menulis kode, meng-*compile*, mencari kesalahan dan memberikan kemudahan dalam membangun aplikasi dekstop.

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

Menurut Pratama (2014:440), XAMPP merupakan “Aplikasi *Web Server* bersifat Instan (siap haji) yang dapat digunakan baik disistem operasi *Linux* maupun dari sistem operasi *windows*”.

Menurut Aryanto (2016:4) XAMPP merupakan “Sebuah aplikasi perangkat lunak pemrograman dan *database* yang di dalamnya terdapat berbagai macam aplikasi pemrograman seperti *Apace HTTP Server*, *MySQL Database*, bahasa pemrograman PHP dan *Perl*”.

Berdasarkan dari teori yang telah dijelaskan sebelumnya disimpulkan bahwa XAMPP (X (Windows/Linux) Apache MySQL PHP Perl) adalah perangkat lunak pemrograman dan *database* yang bersifat instan serta pengoperasian yang sangat mudah.

2.2.3. Pengertian MySQL (*My Structured Query Language*)

Menurut wahana komputer (2010:5) mengatakan “MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan *multy user*”. Adapun dalam pendapat lain “MySQL adalah sebuah manajemen *system database server* yang mampu menangani beberapa *user*, yaitu mampu menangani beberapa intruksi sekaligus dari beberapa *user* dalam satu waktu” (Andi, 2010:9)’

Sedangkan menurut Anhar (2010:21) berpendapat *MySQL (My Structure Query Language)* adalah ”sebuah perangkat lunak sistem manajemen basis data *SQL (Database Manajemen System)* atau *DBMS (Database manajemen System)* dari sekian banyak *DBMS (Database manajemen System)*, seperti *Oracle.MS SQL,Postage SQL*, dan lain-lain”.

Berdasarkan dari penjelasan yang telah dibahas dapat disimpulkan bahwa *MySQL (My Structure Query Language)* perangkat lunak sistem manajemen *database* yang dapat mengirim dan menerima data dengan sangat cepat serta *multy user*.

2.2.4. Pengertian *PHPMyAdmin*

Menurut Putri dalam Hikmah dkk (2015:2) *PHPMyAdmin* merupakan “Aplikasi yang dapat digunakan untuk membuat *Database*, pengguna (*user*), memodifikasi tabel, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan perintah (*commond*) *SQL (Structure Query Language)*”.

Sedangkan menurut Prasetyo (2014:48) mengatakan bahwa *PHPMyAdmin* merupakan “*Tools* berbasis *web* yang berguna untuk mengelola *database MySQL*”.

Berdasarkan dari uraian yang telah dipaparkan sebelumnya, disimpulkan bahwa *PHPMyAdmin* merupakan aplikasi khusus untuk mengolah *database*, pengguna (*user*), memodifikasi tabel dan mengirim *database* dengan cepat dan mudah.

2.2.5. UML (*Unified Modeling Language*)

menurut Sukamto dan Shalahuddin (2014:133), berpendapat bahwa UML (*Unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requerement*, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Sedangkan menurut Mulyani (2016:48) mengatakan UML (*Unified Modeling Language*) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek.

UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya (Sukamto dan Shalahuddin, 2014:155):

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informai yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi

apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

2. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

3. *Class Diagram*

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.

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 menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

2.2.6. Pengujian

Menurut Simarmata (2010:301) berpendapat bahwa Pengujian adalah “Proses eksekusi suatu program untuk menemukan kesalahan”. Adapun

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

Jadi dapat disimpulkan bahwa pengujian adalah suatu proses mengevaluasi program dengan tujuan menemukan kesalahan agar program dapat berjalan sesuai perencanaan.

Menurut Sukanto dan Shalahudin (2014:275) untuk memastikan validasi dalam sebuah pengujian memiliki beberapa pendekatan sebagai berikut:

1. *Black-Box Testing* (Pengujian Kotak Hitam)

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

2. *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. Pengujian kotak putih dilakukan dengan memeriksa logika dari kode program.