

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Dewasa ini program (program komputer) sudah menjadi kebutuhan yang sangat penting bagi perusahaan. Hampir seluruh aspek kehidupan manusia saat ini tidak dapat dilepaskan dari program dan teknologi komputer. Dimulai dari rumah tangga, sekolah, kantor, pertokoan, hingga industri-industri baik industri kecil, sedang maupun besar sudah bergantung kepada program (program komputer).

Program adalah teknik komunikasi standar untuk mengekspresikan instruksi kepada komputer. Layaknya bahasa manusia, setiap bahasa memiliki tata tulis dan aturan tertentu. Program memfasilitasi seorang *programmer* untuk secara spesifik apa yang akan dilakukan oleh program (program komputer) selanjutnya, bagaimana data tersebut disimpan dan dikirim, dan apa yang akan dilakukan apabila terjadi kondisi yang variatif.

2.1.1. Pengertian Program

Menurut Kurniawan (2011:65), “Program adalah kumpulan instruksi yang disusun dengan urutan nalar yang tepat untuk menyelesaikan suatu persoalan”.

Sedangkan menurut Suarga (2009:17), “program adalah susunan instruksi dalam suatu bahasa pemrograman yang menjelaskan kepada mesin (komputer) bagaimana melakukan sebuah proses terhadap data”.

Sehingga dapat disimpulkan juga bahwa sebuah program merupakan himpunan atau kumpulan instruksi tertulis yang dibuat oleh *programmer* atau suatu bagian berupa instruksi atau statement yang tentunya dalam bahasa yang

dimengerti oleh komputer. Instruksi tersebut berfungsi untuk mengatur pekerjaan apa saja yang akan dilakukan oleh komputer agar mendapatkan dan menghasilkan suatu hasil atau keluaran yang diharapkan.

2.1.2. Pengertian Aplikasi

Menurut Hendrayudi (2009:143), “Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus)”.

Sedangkan menurut Zaki dan Smitdev Community (2007:11), “aplikasi (program aplikasi) adalah komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data”.

Berdasarkan uraian diatas, maka aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia.

2.1.3. Pengertian Penerimaan Kas

Menurut Arifin (2007:95), “penerimaan kas atau kas masuk (*cash inflow*) uang yang benar-benar diterima perusahaan”.

Sedangkan menurut Hall (2007:258), “prosedur penerimaan kas adalah sistem *bacth* alami, tidak seperti transaksi penjualan, yang cenderung terjadi terus-menerus dalam satu hari, penerimaan kas merupakan kejadian yang berbeda”.

Jadi dapat disimpulkan bahwa, penerimaan kas adalah suatu kesatuan untuk mengumpulkan, mencatat transaksi yang dapat membantu perusahaan untuk menangani penerimaan kas perusahaan.

2.1.4. Pengertian Pengeluaran Kas

Menurut Hartoko (2011:120), “pengeluaran kas adalah seluruh pengeluaran yang dilakukan dari kas suatu usaha dalam kurun waktu tertentu, misalnya bulanan, semester, dan lain-lain”.

Sedangkan menurut Arifin (2007:100), “pengeluaran kas atau kas keluar (*cash outflow*) merupakan aliran pembayaran tunai yang dilakukan perusahaan, kebalikan dari kas masuk”.

Berdasarkan pengertian pengeluaran kas menurut para ahli di atas, dapat disimpulkan bahwa pengeluaran kas adalah suatu proses yang dilaksanakan untuk melaksanakan pengeluaran kas baik dengan cek maupun uang tunai untuk kegiatan pengeluaran kas perusahaan.

2.2. Peralatan Pendukung (*Tools System*)

2.2.1. Visual Basic

Sejarah Visual Basic diawali dari pengembangan bahasa BASIC (Beginners All-Purpose Symbolic Instruction Code) di Dartmouth College, Amerika Serikat pada awal tahun 1960. Sejak semula Basic dirancang untuk mudah dipelajari pada awal tahun 1975 muncul komputer pribadi (Personal Computer /PC) pertama, bermerek MITS Altair. Namun karena hanya memiliki RAM 4 KB, satu-satunya bahasa yang bisa digunakan untuk pemrograman adalah Assembly, bahasa ini sangat sulit untuk dipahami karena perintahnya mendekati bahasa mesin. Tentu hal ini sangat menghambat perkembangan komputer pribadi. Bill Gate dan Paul Allen melihat potensi yang sangat besar pada komputer pribadi dimasa depan. Keduanya menciptakan BASIC untuk Altair. Ternyata sambutan masyarakat

sangat baik dan akhirnya keduanya mendirikan perusahaan sendiri yang bergerak dibidang *software* yang bernama *Microsoft*.

Pada tahun 1982 IBM/PC diperkenalkan kepada masyarakat, Microsoft pun membuat sistem operasi MS-DOS untuk komputer ini. Dimana didalamnya disertakan pula bahasa Basic yang dikenal dengan sebagai Quick Basic (QBASIC). Zamanpun berganti, pada tahun 1990 era DOS berlalu dan digantikan oleh era Windows.

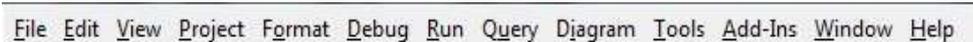
Dari perkembangan ini akhirnya Microsoft membuat Basic versi Windows, bahasa pemrograman Basic dikenal dengan nama Visual Basic. Versi terakhirnya adalah Microsoft Visual Basic 6.0 for Windows 98. Saat ini perkembangan visual basic telah mengarah ke pemrograman .NET , yaitu dikenal dengan nama Visual Basic.NET. (Mesran, 2009:1).

Menurut Amidun (2016:17), “Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows”.

Sedangkan menurut Winarto dan Zaki (2010:1), “Visual Basic adalah bahasa pemrograman klasik, legendaris, dan tiada duanya yang paling banyak dipakai oleh *programmer* dunia”.

Menurut Mesran (2009:4), ada beberapa komponen yang sangat mendukung di dalam suatu program Visual Basic yaitu:

1. Baris Menu



File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

Sumber : Mesran (2009:4)

Gambar II.1 Baris Menu

Baris menu terletak paling atas pada IDE, menu merupakan kumpulan perintah-perintah yang dikelompokkan dalam kriteria operasi yang dihasilkan.

2. Toolbar



Sumber : Mesran (2009:4)

Gambar II.2 Toolbar

Toolbar adalah tombol-tombol yang mewakili suatu perintah tertentu dari Visual Basic. Biasanya tombol-tombol ini merupakan perintah-perintah yang sering digunakan.

3. Form



Sumber : Mesran (2009:6)

Gambar II.3 Form

Form adalah suatu objek yang dipakai sebagai tempat bekerja program aplikasi. Didalam form terdapat garis titik-titik yang disebut grid yang membantu pemakai dalam pengaturan tata letak objek dalam form. Umumnya pada form terdapat garis titik-titik yang disebut dengan grid.

4. ToolBox



Sumber : Mesran (2009:7)

Gambar II.4 ToolBox

Toolbox adalah tempat penyimpanan kontrol yang akan kita gunakan pada program yang di pasangkan pada form.

5. Windows Form Layout

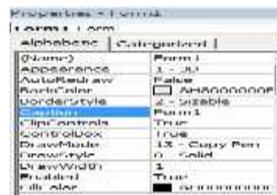


Sumber : Mesran (2009:9)

Gambar II.5 Windows Form Layout

Form layout window adalah jendela yang menggambarkan posisi dari form yang ditampilkan pada layar monitor.

6. Window Property



Sumber : Mesran (2009:9)

Gambar II.6 Window Property

Property digunakan untuk menentukan setting suatu objek. Suatu objek biasanya mempunyai beberapa properti yang dapat diatur langsung dari jendela propertis atau lewat jendela program. Untuk menggunakan properti yaitu click terlebih dahulu kontrol yang ingin kita setting.

7. Project Explorer

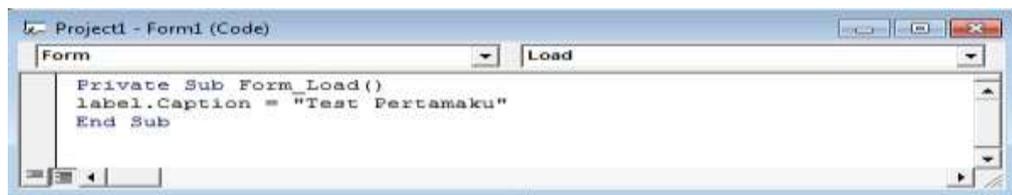


Sumber : Mesran (2009:10)

Gambar II.7 Project Explorer

Project adalah sekumpulan modul. Jadi project atau proyek merupakan program aplikasi itu sendiri. Project disimpan dalam file yang berakhiran.VBP. file ini menyimpan seluruh komponen-komponen program, termasuk pilihan proyek, pilihan environment, pilihan file *EXE* dan segala sesuatu yang berkaitan dengan proyek.

8. Code Window



Sumber : Mesran (2009:10)

Gambar II.8 Code Window

Code window adalah jendela yang berisi kode-kode program yang merupakan instruksi-instruksi program untuk aplikasi Visual Basic.

2.2.2. Microsoft Visual Basic 6.0

Menurut Fauzi dan Amin (2012:1), “Microsoft Visual Basic 6.0 merupakan bahasa pemrograman yang mudah digunakan untuk pengembangan aplikasi, baik aplikasi kecil maupun aplikasi besar”.

Sedangkan menurut Madcoms (2010:2), “Microsoft Visual Basic 6.0 merupakan bahasa pemrograman yang cukup populer dan mudah untuk dipelajari”.

Jadi dapat disimpulkan bahwa, Microsoft Visual Basic 6.0 merupakan bahasa pemrograman yang bekerja dalam ruang lingkup Microsoft Windows”.

2.2.3. Database

Menurut Rosa dan Shalahuddin (2015:43), “sistem *database* adalah Sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Sedangkan menurut Sri Mulyani (2016:156), “*Database* merupakan salah satu komponen terpenting dalam sebuah sistem informasi karena semua informasi yang akan diolah dan dihasilkan tersimpan dalam *database*”.

Jadi *database* adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.

2.2.4. MySQL ODBC Connector

Menurut Saputra (2010:3), “MySQL ODBC Connector adalah *software* kecil dengan fungsi yang besar”.

Sedangkan menurut Fauzi dan Amin (2012:59), “MySQL ODBC Connector (*Open Database Connectivity*) merupakan *application programming interface* (API) database yang khusus digunakan untuk mengakses database relational”.

Jadi dapat disimpulkan bahwa MySQL ODBC Connector adalah suatu kerangka kerja umum untuk mengakses dan mengubah isi *database*.

2.2.5. ODBC

Menurut Andoyono dan Suyono (2016:209), “ODBC adalah program untuk koneksi *database* secara lokal maupun *remote* (jarak jauh), dan juga dapat menangani berbagai *database* dengan format berbeda, dengan catatan *driver*”.

Sedangkan menurut Fauzi dan Amin (2012:59), “*Open Database Connectivity (ODBC)* merupakan *Application programming Inteface (API)* *database* yang khusus digunakan untuk mengakses *database* relasional”.

Menurut Fauzi dan Amin (2012:59) pada ODBC terdapat dua hal penting yang berkaitan dengan koneksi *database* yaitu:

1. *Database Driver*

Database Driver adalah *software* atau *driver* yang berfungsi untuk membuat koneksi ke *database* yang sesuai dengan *database* tersebut. Setiap *database* memiliki driver yang spesifik, sehingga dalam sebuah ODBC data berisi lebih dari satu *database driver*.

2. *Data Source Name (DSN)*

Data Source Name (DSN) adalah nama atau alias dari *database* yang digunakan oleh program, yang koneksinya melalui ODBC.

Menurut Fauzi dan Amin (2012:60) ada beberapa jenis *data source name* yang akan menentukan otoritas pemakai *database*, yaitu:

1. *User DSN*

User DSN bersifat privat bagi *user* pembuatnya, artinya DSN jenis ini hanya dapat dipergunakan atau tampak oleh *user* yang membuatnya (yaitu *user* yang *login* dengan username pembuat *user DSN* tersebut), yang perlu menjadi catatan adalah bahwa sifat *user DSN* hanya berjalan pada windows NT.

2. *System DSN*

System DSN bersifat *public* dalam sebuah komputer, artinya semua *user* yang menggunakan komputer dimana sistem *DSN* diinstal dapat menggunakannya walaupun bukan pembuatnya.

3. *File DSN*

File DSN adalah satu-satunya yang bisa di-*share* dan dapat dipergunakan komputer lain yang memiliki *driver database* yang sama, dengan demikian *setting* untuk koneksi *database* cukup dilakukan pada satu komputer saja dan komputer yang lain cukup *me-load*-nya saja. *File DSN* ini umumnya dipergunakan untuk aplikasi *multiuser* pada suatu LAN.

2.2.6. WampServer

Menurut Meissa (2009:17), “WampServer adalah singkatan dari Windows, Apache, Mysql dan PHP”.

Menurut Zaenal (2011:10), “WampServer adalah aplikasi terpaket yang berisi PHP, MySQL, dan Apache untuk menyimpan dan menerjemahkan database menjadi sebuah halaman website”.

Jadi dapat disimpulkan bahwa WampServer adalah paket web server yang bekerja secara indenpenden.

2.2.7. Crystal Report

Menurut Junindar (2008:132), “Crystal Report adalah tool untuk membuat laporan yang mempunyai kemampuan andal dalam menampilkan laporan, karena didalamnya banyak sekali fitur-fitur untuk memudahkan kita dalam membuat laporan”.

Sedangkan menurut Amidun (2016:158), “Crystal Report adalah suatu form khusus berbentuk seperti lembaran format naskah yang ingin dicetak”.

Jadi dapat disimpulkan bahwa, Crystal Report adalah salah satu software yang dapat untuk membuat laporan dalam pemrograman.

2.2.8. UML (*Unified Modeling Language*)

Menurut Rosa dan Shalahuddin (2015:137), “UML merupakan bahasa visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak”.

Sedangkan menurut Nugroho (2010:6), “UML (*Unified Modeling Language*) adalah bahasa pemrograman untuk sistem atau perangkat lunak yang berparadigma berorientasi objek”.

Berdasarkan pengertian uml dari para ahli, maka dapat disimpulkan bahwa UML adalah himpunan struktur dan teknik untuk pemodelan desain program serta aplikasinya.

Beberapa diagram UML yang digunakan didalam laporan Tugas Akhir ini, antara lain:

1. *Use Case Diagram*

Menurut Rama dan Jones (2008:329), “*Use Case* adalah urutan langkah-langkah yang terjadi ketika ‘pelaku’ sedang berinteraksi dengan sistem untuk suatu tujuan tertentu”.

Sedangkan menurut Rosa dan Shalahuddin (2015:155), “*Use Case* atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

Syarat penanaman pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. (Rosa dan Shalahuddin, 2015:156).

a. Nama *Use Case*

Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*.

b. Aktor

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, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

c. Asosiasi

Komunikasi antara aktor dan nama *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.

d. Ekstensi

Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu.

e. Generalisasi

Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

2. Activity Diagram

Menurut Widodo dan Hertawati (2011:11), “diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem”.

Menurut Rosa 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”.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

a. Status Awal

Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

b. Aktivitas

Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

c. Percabangan

Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

d. Penggabungan

Asosiasi penggabungan dimana ada pilihan aktivitas penggabungan lebih dari satu.

e. Status Akhir

Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir.

2.2.9. Entity Relationship Diagram (ERD)

Menurut Rosa dan Shalahuddin (2015:53), “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional”.

Sedangkan menurut Yanto (2016:32), “ERD adalah suatu diagram untuk menggambarkan desain koseptual dari model konseptual suatu basis data relasional”.

Berdasarkan pengertian ERD diatas dapat disimpulkan bahwa ERD adalah satu model yang digunakan untuk mendesain *database* dengan tujuan menggambarkan data yang berelasi pada sebuah *database*.

2.2.10. Logical Relational Structure (LRS)

Menurut Dhanta (2009:193), "*logical Relational Structure (LRS)* adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas menentukan kardinalitas, jumlah tabel dan *foreign key*".

Menurut Friyadie (2007:13), "LRS merupakan hasil dari pemodelan Entity Relation Ship (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas".

Dalam pembuatan LRS terdapat 3 hal (Friyadie, 2007:13), yaitu:

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.