

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Konsep pemrograman terstruktur memegang peran penting dalam merancang. Menyusun dan mengembangkan suatu program. Program disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa merancang, menyusun dan memelihara serta mengembangkan suatu program untuk menyelesaikan masalah dan diimplementasikan dengan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer.

A. Program

Menurut (Sahyar, 2016:2) “Program adalah perintah-perintah atau instruksi yang disusun berdasarkan algoritma dengan menggunakan Bahasa pemrograman untuk penyelesaian suatu masalah”.

B. Bahasa Pemrograman Java

Menurut (Sahyar, 2016:2) “Bahasa Pemrograman adalah perangkat lunak yang digunakan untuk menerjemahkan atau menuliskan algoritma dalam bentuk teks perintah-perintah yang dapat di mengerti oleh computer untuk menyelesaikan suatu masalah”

Menurut (Ems, 2015:5) “Java adalah bahasa pemrograman yang dirancang tidak tergantung kepada platform alias platform independent, sehingga program yang ditulis dibahasa pemrograman java idealnya harus bisa dijalankan disemua sistem operasi”

Menurut (Nofriadi, 2018:1) “Java merupakan bahasa pemrograman yang bisa dijalankan di berbagai jenis sistem operasi sehingga dikenal juga bahasa pemrograman *multiplatform*, yang bersifat pemrograman berorientasi objek, dan memiliki *library* yang lengkap”.

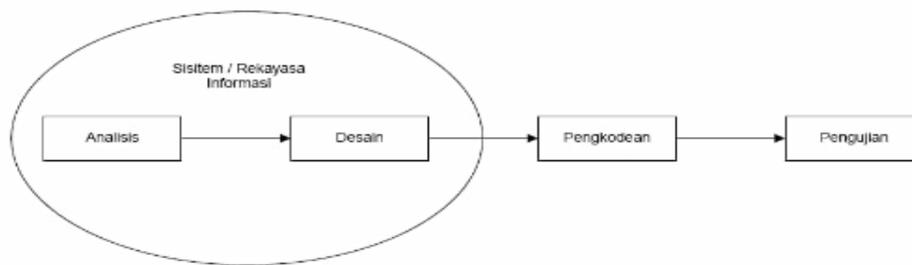
C. Basis Data

Menurut (Pamungkas, 2017:2) “Basis data sebagai sekumpulan data yang disusun dalam bentuk beberapa tabel yang saling memiliki relasi maupun berdiri sendiri”.

Menurut (Sukamto & Shalahuddin, 2015:43) “Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

D. Model Pengembang Perangkat Lunak

Menurut (Sukamto & Shalahuddin, 2015:28) Model SDLC air terjun (*waterfall*) sering disebut juga model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau turut dimulai dari analisis, desain, pengodean, pengujian, dan tahapan pendukung (*support*). Berikut adalah gambar model air terjun:



Sumber: (Sukamto & Shalahuddin, 2015:29)

Gambar II.1
Ilustrasi Model *Waterfall*

1. Analisa Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara *intensif* untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*).

5. Pendukung (*support*) atau pemeliharaan (*maintanance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

2.2. Tools Program

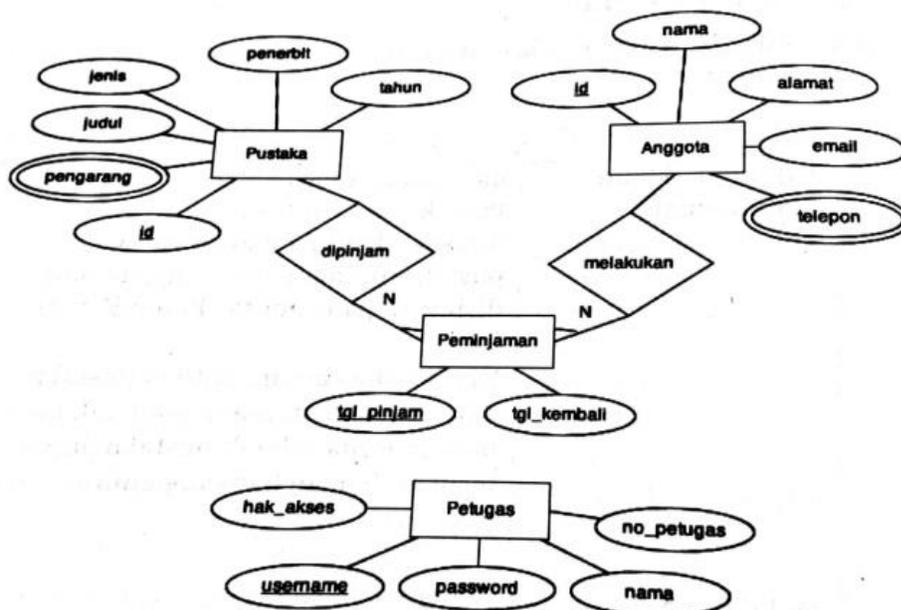
Dalam penulisan Tugas Akhir ini Tools Program yang digunakan oleh penulis meliputi *Entity Relationship Diagram*, pengkodean, HIPO (*Hierarchy Input Proses Output*), Diagram Alir (*Flowchart*), *Implementasi* dan pengujian unit.

A. Enterprise Relationship Diagram

1. Pengertian *Entity Relationship Diagram*

Menurut (Mulyani, 2016:100) “ERD merupakan *tools* yang digunakan untuk memodelkan struktur data dengan menggambarkan entitas dan hubungan antara entitas (*Relatiohsip*)”

Menurut (Sukamto & Shalahuddin, 2015:50) *Entity Relationship Diagram* atau ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODMS maka perancangan basis data tidak diperlukan menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Elis), notasi Crow’s Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.



Gambar 16 Diagram ER Studi Kasus

Sumber: (Sukamto & Shalahuddin, 2015:58)

Gambar II.2 Entity Relationship Diagram

2. Komponen ERD

Ada tiga komponen yang digunakan pada ERD (*Entity Relationship Diagram*) dengan notasi Chen menurut (Sukamto & Shalahuddin, 2015:50):

a. Entitas (*Entity*)

Entitas adalah suatu objek di dunia nyata yang dapat dibedakan dengan objek lainnya. Objek tertentu dapat berupa orang, benda ataupun hal lainnya. Dari jenis entitas terbagi atas dua, yaitu:

1. Entitas Kuat (*Strong Entity*)

Entitas kuat adalah entitas yang dapat berdiri sendiri tidak bergantung pada entitas lainnya. Entitas kuat memiliki *atribut key*, dan entitas kuat digambarkan sebagai kotak persegi panjang bergaris tunggal.

2. Entitas Lemah (*Work Entity*)

Entitas lemah adalah entitas yang tidak dapat berdiri sendiri. Entitas lemah merupakan hasil dari pembentukan entitas kuat, entitas lemah tidak memiliki *atribut key* dan entitas lemah digambarkan sebagai kotak persegi panjang.

b. Atribut (*Attribute*)

Atribut merupakan semua informasi yang berkaitan dengan entitas. Atribut sering dikenal dengan *property* dari suatu entitas atau objek. Atribut digambarkan dalam bentuk lingkaran *elips*. Macam-macam atribut yaitu:

1. Atribut Sederhana (*Simple Attribute*)

Atribut sederhana adalah atribut yang nilainya tidak dapat dibagi lagi menjadi banyak yang lebih kecil.

2. Atribut Komposit (*Composite Attribute*)

Atribut komposit adalah atribut gabungan yang nilainya dapat dipecah menjadi bagian yang lebih kecil.

3. Atribut Bernilai Tunggal (*Single Values Attribute*)

Atribut bernilai tunggal adalah jenis atribut yang bernilai satu dari suatu entitas.

4. Atribut Bernilai Banyak (*Multivalues Attribute*)

Atribut bernilai banyak adalah jenis atribut yang nilainya lebih dari satu dalam suatu entitas tertentu.

5. Atribut Turunan (*Derived Attribute*)

Atribut turunan adalah jenis atribut yang nilainya diperoleh dari atribut yang lain.

6. Atribut Identitas (*Key Attribute*)

Atribut identitas adalah atribut yang dijadikan sebagai kunci pada suatu *table*. Sifat atribut identitas ini unik, tidak ada yang menyamai.

7. Atribut Kunci Primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan. Atribut terdiri dari beberapa jenis yaitu:

a. *Super Key*

Super key adalah suatu atribut data kumpulan atribut yang secara unik mengidentifikasi sebuah garis didalam relasi atau himpunan dari satu atau lebih entitas yang dapat digunakan untuk mengidentifikasi secara unik sebuah entitas dalam set entitas.

b. *Candidate Key*

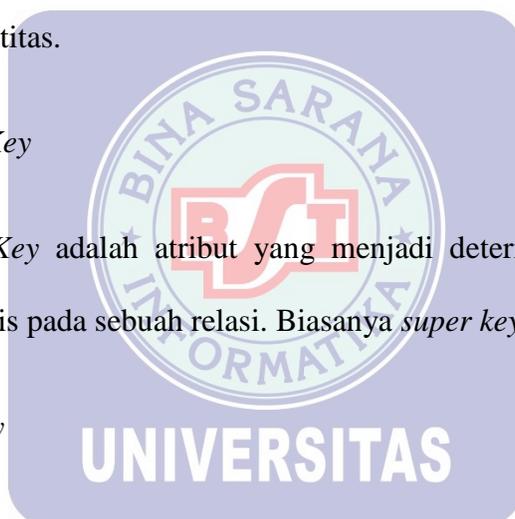
Candidate Key adalah atribut yang menjadi determinan yang dapat dijadikan identitas baris pada sebuah relasi. Biasanya *super key* minimum.

c. *Primary Key*

Primary Key adalah kandidat *key* yang dipilih untuk mengidentifikasi basis data secara unik dalam relasi.

d. *Alternative Key*

Alternative key adalah *candidate key* yang tidak dipilih sebagai *primary key* atau atribut untuk menggantikan kunci utama.



e. *Foreign Key*

Foreign key adalah atribut dengan domain yang sama yang menjadi kunci utama sebuah relasi, tetapi pada relasi lain atribut tersebut sebagai atribut biasa.

f. *Composite Key*

Composite key adalah kunci yang terdiri dari dua atribut atau lebih. Atribut-atribut tersebut jika berdiri sendiri tidak menjadi identitas baris, tetapi bila dirangkaikan menjadi satu kesatuan akan dapat mengidentifikasi secara unik.

c. Relasi (*Relationship*)

Relasi merupakan notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan. Aturan penggambaran relasi adalah sebagai berikut:

1. Relasi dinyatakan dengan simbol belah ketupat.
2. Nama relasi dituliskan didalam simbol belah ketupat.
3. Nama relasi berupa kata kerja aktif.
4. Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan makna dengan jelas.

2. Logical Record Structure

Menurut Tabrani dalam Kuryanti (Kuryanti, 2016) “*Logical Record Structure (LRS)* dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak persegi panjang dan dengan nama yang unik”.

B. Pengkodean

Menurut (Kristanto, 2018:106) “Pengkodean adalah suatu tahap dari analisa kebutuhan sistem dan desain sistem yang dituliskan dalam suatu bahasa pemrograman komputer tertentu yang biasanya oleh pabrik komputer sudah ditentukan spesifikasinya”.

Macam-macam tipe kode yang dapat digunakan antara lain menurut (Shatu, 2016:108), yaitu:

1. Kode Urut

Kode yang disusun urut nomor. Agar setiap kode mempunyai jumlah angka (digit) yang sama perlu direncanakan dulu jumlah digitnya, misal jumlah digitnya sebanyak empat angka maka kodenya akan dimulai dengan 0001 dan diakhiri 9999. Kode urut ini sederhana, tetapi tidak memenuhi persyaratan fleksibilitas.

2. Kode Kelompok

Kode kelompok membagi data ke dalam kelompok tertentu. Tiap kelompok akan diberi kode dengan angka, sehingga masing-masing posisi angka kode mempunyai arti.

3. Kode Blok

Dalam kode blok, setiap kelompok data diberi kode dalam blok nomor tertentu. Cara pemberian kode ini dapat memenuhi persyaratan *fleksibilitas* sehingga dapat digunakan untuk pemberian kode pada rekening.

4. Kode Desimal

Setiap kelompok data akan diberi kode 0 sampai 9. Oleh karenanya pengelompokkan data harus dilakukan maksimum dalam sepuluh kelompok. Agar kode decimal ini dapat digunakan untuk pengelompokkan data yang luas, dapat disusun kelompok-kelompok yang bertingkat.

5. Kode *Mnemonic*

Merupakan singkatan dari karakteristik data. Misal pabrik sepatu, persediaan sepatu pria ukuran besar dapat dibuatkan kode SPB (sepatu pria besar). Bisa juga kode ini disusun dengan kombinasi huruf dan angka, misal sepatu pria dengan nomor 42 diberi kode SP42. Sebaiknya kode mnemonic ini digunakan bila data atau elemennya (itemnya) tidak terlalu banyak, sehingga tidak menyulitkan pemakaiannya. Bila terlalu sering perubahan itemnya dan banyak data maka pemakaian kode akan sulit mengingatnya.

6. Kode Baru

Digunakan untuk *industry* makanan dan minuman diluar negeri (misal: USA) yang menggunakan *Universal Product Code* (UPC). Tiap pengusaha makanan dan minuman.

C. HIPO (*Hierarchy Input Proses Output*)

1. Pengertian HIPO

Menurut (Suprpto, 2018:167) HIPO (*Hierarchy Input Proses Output*) merupakan teknik dan alat yang digunakan untuk merancang dan mendokumentasikan program komputer. Hipo juga dapat membantu perancangan Program untuk mengevaluasi dan menyempurnakan rancangan program dan mendeteksi kesalahan sebelum implementasi.

2. Tingkatan Diagram Hipo

Fungsi-fungsi dari sistem digambarkan oleh HIPO dalam tiga tingkatan yaitu:

a. *Visual Table of Content* (VTOC)

Diagram ini menggambarkan hubungan dari fungsi-fungsi di sistem secara berjenjang.

b. *Overview Diagram*

Overview Diagram menunjukkan secara garis besar hubungan dari input, proses, output. Bagian input menunjukkan item-item data yang akan digunakan oleh bagian proses. Bagian proses berisi jumlah langkah-langkah yang menggambarkan kerja

dan fungsi. Bagian output berisi item-item data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses.

c. *Detail Diagrams*

Detail Diagram merupakan diagram tingkat yang paling rendah di diagram HIPO. Diagram ini berisi dengan elemen-elemen dasar dari paket yang menggambarkan secara rinci kerja dan fungsi.

D. Diagram Alir (*Flowchart*)

1. Pengertian *Flowchart*

Menurut (Suprpto, 2018:66) “*Flowchart* merupakan model untuk merepresentasikan pengolahan data informasi dan dokumen yang terjadi suatu proses, baik proses bisnis maupun proses pada suatu program”.

2. Bentuk *Flowchart*

a. Program *Flowchart*

Simbol-simbol *flowchart* yang digunakan untuk menggambarkan logik dari pemrosesan terhadap data.

b. Sistem *Flowchart*

Simbol-simbol peralatan sistem komputer yang digunakan untuk menyatakan proses pengolahan data.

3. Teknik Pembuatan

a. *General Way*

Teknik pembuatan *flowchart* yang digunakan pada penyusun logika dalam suatu program pengulangan proses secara tidak langsung (*Non-Direct Loop*).

b. *Internal Way*

Teknik pembuatan *flowchart* yang digunakan pada penyusunan logika program yang cepat dan pengulangan proses yang terjadi bersifat langsung (*Direct-Loop*).

E. Implementasi Dan Pengujian Unit

Menurut (Indrajani, 2017:11) “*Implementation* (Implementasi) merupakan realisasi fisik dari basis data dan desain aplikasi”.

Menurut (Sukamto & Shalahuddin, 2015:272) Pengujian merupakan satu set aktifitas yang direncanakan secara sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji spesifik dan metode pengujian.

Menurut (Sukamto & Shalahuddin, 2015:275) *Black Box Testing* (pengujian kotak hitam) merupakan 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.

