

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Seiring dengan perkembangan zaman dimasa sekarang ini, sistem adalah salah satu faktor yang mempengaruhi dalam kinerja suatu perusahaan. Dengan adanya sistem perusahaan terbantu dalam menjalankan bisnis di perusahaan sehingga dapat mempermudah serta efisien dalam pekerjaan dan pengolahan data yang baik.

Awalnya sistem komputer dibuat hanya untuk mempercepat sebuah proses pekerjaan. Seiring berjalannya waktu dan tingkat kemampuan sumber daya manusia baik dari sisi teknologi maupun pengguna serta bisnis, pembuatan dan penggunaan sistem komputer semakin lama semakin meningkat. Sistem komputer tidak lagi hanya untuk mempercepat tetapi juga mempermudah pembuatan laporan, mempermudah proses pengecekan berdasarkan alur data, dan mengunci kemungkinan penyelewengan.

Dunia bisnis melaju dengan begitu pesat hampir dari perusahaan yang bersekala kecil maupun besar telah menggunakan teknologi informasi guna memberikan peningkatan terhadap layanan bisnis yang mereka kelola.

Pelumas ada dua macam yakni minyak pelumas dan gemuk pelumas. Gemuk adalah pelumas dalam bentuk semi fluid, digunakan untuk melumsmesin yang bekerja pada putaran rendah tapi beban besar. Kebutuhan gemuk dunia lebih dari 1,5 milyar ton setiap tahun dan terus meningkat.

Minyak gemuk merupakan pelumas yang berbentuk semi fluid. Gemuk diperlukan untuk menjalankan mesin-mesin industri dan mesin kendaraan bermotor. Gemuk digunakan pada bagian mesin yang bergesekan satu dengan yang lainnya yang tidak dapat dilumasi dengan minyak pelumas biasa. Penggunaan minyak gemuk antara lain untuk bantalan luncur dan bantalan peluru pada mesin industri, pada laker, pada gigi dan lain-lain.

Beberapa pelayanan penjualan yang bergerak dibidang produksi dan penjualan minyak gemuk masih menggunakan program MS Excel dan manual, sehingga kurang efesienya pengolahan data penjualan dan pemesanan produk yang dilakukan oleh konsumen. maka dari itu dibutuhkan program yang dapat memberi kemudahan dalam pengelolaan data serta informasi atas transaksi penjualan minyak gemuk.

Program yang akan dibuat meliputi dari program untuk pembelian minyak gemuk hingga laporan dalam satu program, sehingga mempermudah admin dalam mengeluarkan *output* serta membuat laporan dari hasil penjualan yang dilakukan. Program penjualan minyak gemuk ini dibuat atau dirancang sedemikian rupa hingga menghasilkan program yang banyak memberi kemudahan dalam pembelian serta informasi penjualan.

1.2. Tujuan dan Manfaat

Tujuan dari penulisan tugas akhir ini adalah

1. Untuk mengetahui bagaimana pengolahan transaksi penjualan dan pengolahan data-data penjualan minyak gemuk
2. Membuat program penjualan minyak gemuk untuk menghasilkan transaksi yang cepat, tepat dan akurat.

3. Agar program penjualan dapat dimanfaatkan dan digunakan oleh instansi sebagai program penjualan minyak gemuk.

Sedang manfaat penulisan tugas akhir adalah sebagai berikut:

1. Manfaat untuk penulis

Sebagai salah satu syarat kelulusan Program Diploma Tiga (DIII) Program Sistem Informasi di Universitas Bina Sarana Informatika.

2. Manfaat untuk objek penelitian

- a. Sebagai bahan evaluasi terhadap sistem yang sedang berjalan.
- b. Menghasilkan suatu informasi yang dapat membantu dan memudahkan proses transaksi penjualan minyak gemuk.
- c. Menghasilkan suatu program yang dapat mempermudah dalam pengolahan data serta penjualan minyak gemuk.

3. Manfaat untuk pembaca

- a. Memberi pemahaman mengenai konsep perancangan sistem penjualan dan transaksi sehingga lebih optimal.
- b. Diharapkan dapat menjadi bahan referensi baru mengenai perancangan program penjualan minyak gemuk sehingga lebih optimal.

1.3. Metode Penelitian

Menurut (A.S & Shalahuddin, 2015) menjelaskan bahwa “model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linear (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Berikut adalah penjelasannya:

1. Analisis kebutuhan perangkat lunak

Pada tahap ini untuk menganalisa kebutuhan perangkat lunak dilakukan dengan Proses pengumpulan data sesuai kebutuhan untuk memspesifikasikan kebutuhan perangkat lunak agar dapat dipahami program seperti apa yang dibutuhkan oleh *user*. Perangkat lunak pada tahap analisa yang dibutuhkan dalam pembuatan program sistem penjualan minyak gemuk yaitu *user*, data pelanggan, data produk, transaksi.

2. Desain

Desain perangkat lunak adalah proses langkah yang fokus pada *desain* pembuatan program perangkat lunak dengan menggunakan *Database Phpmyadmin*, termasuk penggunaa struktur data yang sesuai dengan kebutuhan program, arsitektur perangkat lunak, representasi antar muka dan prosedur pengkodean menggunakan *Java*. Tahap ini mentranslasikan kebutuhan perangkat lunak dari tahap analisis kebutuhan. *Desain* perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

pada tahapan ini *desain* harus ditranslasikan kedalam perangkat lunak. Sehingga menghasilkan program komputer sesuai dengan rancangan yang dibuat. Penullis menggunakan *Java* sebagai pembuatan kode program.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Penulis menggunakan *black box* untuk pengujian perangkat lunak dengan menguji satu persatu setiap kondisi. Hal

ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

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

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. Untuk tahap pemeliharaan dapat dilakukan dengan cara *back up* data secara berkala. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

A. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan penulis dalam melakukan pengumpulan data untuk pembuatan TA adalah :

1. Observasi

Penulis melakukan Observasi untuk mengetahui lebih jelas proses penjualan minyak gemuk, proses pencatatan pembelian atau pemesanan dan pembuatan laporan persediaan dan penjualan minyak gemuk.

Penulis melakukan observasi berfokus pada bagian transaksi penjualan minyak gemuk, sehingga mendapat informasi dengan jelas melalui karyawan pada bidang khusus dalam melakukan penjualan minyak gemuk.

2. Wawancara

Teknik pengumpulan data dengan wawancara adalah teknik yang dilakukan dengan mengajukan beberapa pertanyaan secara langsung yaitu dengan bertatap muka dengan narasumber.

Pengumpulan data dengan wawancara dilakukan dengan mengajukan beberapa pertanyaan terkait yang diperlukan untuk pemesanan, proses pembelian minyak gemuk, *output* yang dihasilkan, dan pembuatan laporan penjualan.

3. Studi Pustaka

Dalam metode ini penulis memperoleh data dengan cara mengumpulkan data melalui buku bacaan, internet, e-jurnal, dan *e-library* yang berhubungan dengan masalah yang dibahas dengan tugas akhir ini.

1.4. Ruang Lingkup

Berdasarkan observasi yang dilakukan oleh penulis, maka penulis membatasi masalah yang akan dibahas dalam ruang lingkup mengenai perancangan program penjualan minyak gemuk, dan pada pembahasan ini penulis berfokus pada proses pengolahan data minyak gemuk, pengolahan data *user*, pengolahan data pelanggan, pengolahan data produk, pengolahan data transaksi, hingga laporan data pelanggan, produk dan transaksi berdasarkan bulan serta tahun.

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

A. Program

Menurut (Kadir, 2017) “Perangkat lunak adalah kumpulan instruksi yang ditunjukkan kepada komputer”. Istilah program dan aplikasi lebih sering disebut untuk menyatakan perangkat lunak. Di kalangan profesional teknologi informasi, istilah program bisa digunakan untuk menyatakan hasil karya mereka berupa instruksi-instruksi untuk mengendalikan komputer. Disini pemakai, hal itu disebut aplikasi. Jadi istilah yang digunakan untuk menyatakan hal yang sama lebih ditentukan oleh masalah persepsi.

Terkait dengan program, terdapat istilah pemogram dan pemograman. Pembuat program dinamakan pemrogram (*programer*). Tugasnya adalah menulis program dan memastikan bahwa program sesuai dengan spesifikasi yang dikehendaki, adapun yang dinamakan pemograman adalah proses untuk menyelesaikan masalah dalam bentuk langkah-langkah penyelesaian yang dapat dikerjakan oleh komputer (yang disebut algoritma) hingga ke penerjemahan kode dalam suatu bahasa pemograman, sehingga masalah tersebut benar-benar bisa dieksekusi oleh komputer.

B. Bahasa Pemograman

Komputer bekerja atas dasar kode biner atau kode yang mempunyai dua keadaan hingga berupa 0 dan 1. Jika dinyatakan dalam keadaan lampau, kode 0 menyatakan keadaan padam dan kode 1 menyatakan keadaan menyala. Atas dasar inilah, program

pada masa awal terciptanya komputer dengan menggunakan bahasa pemrograman yang berbasis pada kode biner dan dinamakan bahasa mesin.

Bahasa mesin dan bahasa rakitan tergolong sebagai bahasa beraras rendah (low level language), yang lebih berorientasi pada mesin. Artinya pemrograman harus benar-benar memahami hal-hal mengenai mesin bersangkutan untuk dapat menuliskan program. Untuk mengatasi kelemahan ini, muncul bahasa beraras tinggi (*high level language*). C, C++, Java, Python, dan Ruby termasuk kategori bahasa beraras tinggi.

Program, apapun bentuknya dibuat menggunakan bahasa pemrograman. Kode yang digunakan untuk menyusun program biasa disebut kode program atau kode sumber (*source code*).

Menurut (Kadir, 2017) “Java adalah bahasa pemrograman yang dapat dijalankan diberbagai komputer, termasuk telepon genggam”. Dikembangkan oleh Sun Microsystems dan dirilis 1995. Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan java tidak hanya berfokus pada suatu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

C. Basis Data

Basis data yang juga dikenal dengan *database* yang merupakan catatan atas kumpulan fakta yang mewakili suatu objek. Data memiliki ciri bersifat mentah dan tidak memiliki konteks. Sedangkan basis atau *base* dapat diartikan sebagai markas, tempat berkumpul dari suatu objek atau representasi objek. Berikut merupakan beberapa definisi basis data :

Basis data didefinisikan menurut Chou (Jayanti & Sumiar, 2018) “ Basis data sebagai kumpulan informasi bermanfaat yang diorganisasikan kedalam tata cara yang

Berdasarkan basis data dapat didefinisikan sebagai sekumpulan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai didalam suatu organisasi. Dalam implementasinya, untuk memudahkan dalam pengaksesan data, data disusun suatu struktur logis yang menjelaskan bahwa :

- a. Kumpulan tabel menyusun basis data.
- b. Tabel tersusun atas sejumlah record
- c. Sebuah record mengandung sejumlah *field*.
- d. Sebuah *field* disimpan dalam bentuk kumpulan bit.

Tujuan dalam merancang basis data adalah

1. Kecepatan dan kemudahan (*speed*)

Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan cepat dan mudah.

2. Efisiensi ruang penyimpanan (*space*)

Penggunaan ruang penyimpanan didalam basis data dilakukan untuk mengurangi jumlah *redundancy* (pengulangan) data, baik dengan melakukan penerapan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antar kelompok data yang saling berhubungan.

3. Keakuratan (*accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan / batasan tipe data, domain data, keunikan data, dan sebagainya, yang diterapkan dalam basis data, sangat berguna untuk menentukan ketidakakuratan pemasukan atau penyimpanan data.

4. Ketersediaan (*availability*)

Pertumbuhan data (baik dari jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar

5. Kelengkapan (*completeness*)

Lengkap atau tidaknya suatu data yang dikelola bersifat relatif, baik terhadap kebutuhan pemakai maupun terhadap waktu. Dalam basis data, struktur dari basis data harus disimpan.

6. Keamanan (*security*)

Sistem keamanan digunakan untuk dapat menentukan siapa saja yang boleh menggunakan basis data dan menentukan jenis operasi apa saja yang boleh dilakukan.

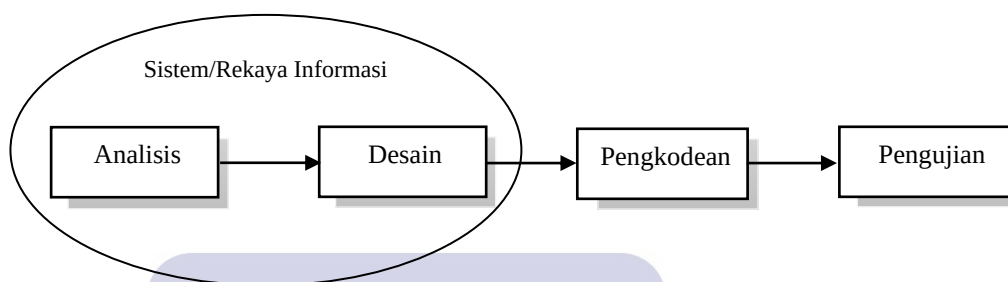
7. Kebersamaan (*shareability*)

Pemakai basis data sering kali tidak terbatas hanya pada suatu pemakainya saja atau oleh suatu sistem aplikasi saja. Basis data yang dikelola oleh sistem yang mendukung lingkungan multiuser, akan dapat memenuhi kebutuhan ini, tetapi dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat bersamaan).

D. Model Pengembangan Perangkat Lunak

Model pengembangan sistem yang digunakan dalam penulisan ini adalah metode pengembangan sistem dengan *waterfall*. Menurut (A.S & Shalahuddin, 2015) mengemukakan bahwa “Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak

secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisa, desain, pengkodean, pengujian, dan tahap pendukungan (*suppport*)". Berikut adalah gambar model air terjun :



Sumber : A.S & Shalahuddin, 2016

Gambar II.1. Ilustrasi Model Waterfall

1. Analisis 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*. Spesifikasi perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. *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 pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program 12 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 segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.
5. Pendukung (*support*) atau pemeliharaan (*maintenance*) Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak yang baru.

2.2. Tools Program

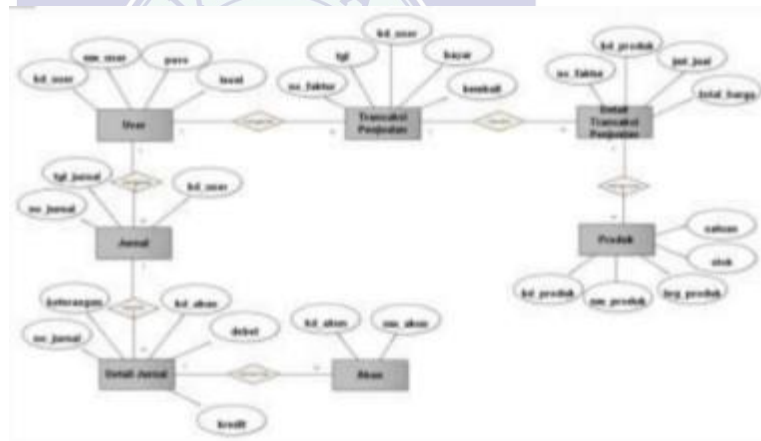
A. *Enterprise relationship diagram (ERD)*

1. Pengertian dan komponen *Enterprise Relation-ship Diagram (ERD)*

Menurut (Trisyanto, 2017) “*ERD* adalah gambaran atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis”. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data. Pada akhirnya *ERD* bisa juga digunakan untuk menunjukkan aturan-aturan bisnis yang ada pada sistem informasi yang dibangun. bagaimana

menggunakan *ERD* untuk menunjukkan aturan bisnis? Ada beberapa poin yang bisa dilihat untuk menjawab pertanyaan ini :

- Aturan bisnis adalah batasan yang harus diikuti ketika sistem beroperasi.
- Simbol *ERD* hanya menunjukkan satu *instance* dari entitas harus ada sebelum *instance* dari suatu entitas.
- Simbol *ERD* dapat menunjukkan ketika salah satu *instance* dari suatu entitas dapat direlasasikan dengan suatu anggota atau lebih dari entitas lainnya.
- Simbol *ERD* juga menunjukkan ketika eksistensinya dari suatu *instance* dalam suatu I adalah opsional untuk sebuah relasi dengan *instance* lain dari suatu entitas



Sumber : (Mesran, 2019)

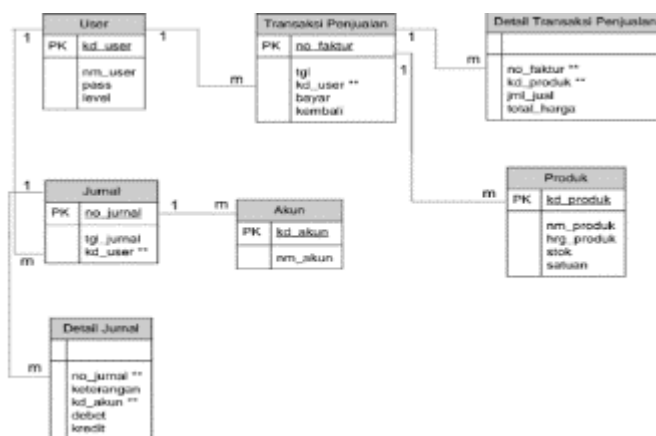
Gambar II.2 Enterprise relationship diagram (ERD)

2. LRS (*logical Record Structure*)

Menurut Deni Kuswono (Kadir, 2017)(Setyawati, 2019) “Logical record structure berasal dari setiap entity yang diubah ke dalam bentuk sebuah kotak dengan nama entity berada di luar kotak dan atribut berada di dalam

kotak”. Untuk menentukan kardinalitas, jumlah table dan Foreign Key sebagai berikut:

- One-to-one Satu entitas berhubungan dengan paling banyak satu entitas lain.
- One-to-many
Satu entitas dapat berhubungan dengan lebih dari satu entitas lain.
- Many-to-many
Beberapa entitas dapat berhubungan dengan beberapa entitas lain.



Sumber : Jurnal MIB Volume 3 No 1 Januari 2019

Gambar II.3 Logical Record Structure (LRS)

B. Pengkodean

Nur, 2016 “Pengkodean yaitu melakukan penerapan hasil rancangan ke dalam bentuk yang dapat dibaca dan di mengerti oleh komputer. Pada tahap ini hasil dari peran-cangan mulai diterjemahkan ke dalam bahasa mesin melalui bahasa pemrograman.

C. HIPO (Hierarki Input Proses Output)

1. Pengertian HIPO

Menurut (Trisyanto, 2017) HIPO (*Hierarchy Plus Input - proses – output*) merupakan metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program. HIPO dapat digunakan sebagai

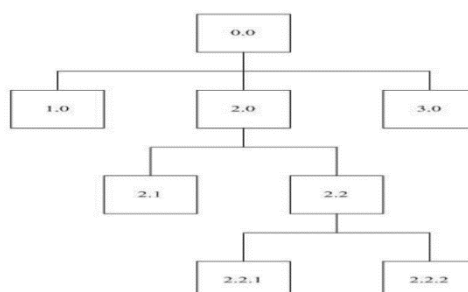
alat pengembangan sistem dan teknik dokumentasi program dan penggunaan HIPO ini mempunyai sasaran utama sebagai berikut:

- a. Untuk menyediakan suatu struktur guna memahami fungsi dari sistem.
 - b. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukan perintah-perintah program yang digunakan untuk melaksanakan fungsi tersebut.
 - c. Untuk menyediakan penjelasan yang jelas dari input yang harus digunakan dan *output* yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari diagram-diagram HIPO.
 - d. Untuk menyediakan *output* yang tepat sesuai dengan kebutuhan pemakai.
2. Tingkatan Diagram HIPO

Fungsi-fungsi dari sistem digambarkan oleh HIPO dalam tiga tingkatan. Untuk masing-masing tingkatan digambarkan dalam bentuk diagram tersendiri. Dengan demikian HIPO menggunakan tiga macam diagram untuk masing-masing tingkatannya, yaitu sebagai berikut.

- a. *Visual Table of Contents* (VTOC)

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



Sumber : Trisyanto, 2017

Gambar II.4. HIPO

b. Overview Diagrams

Overview diagrams menunjukkan secara garis besar hubungan dari input, proses, dan output. Bagian *input* merupakan item-item data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *output* berisi dengan *item-item* data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses.

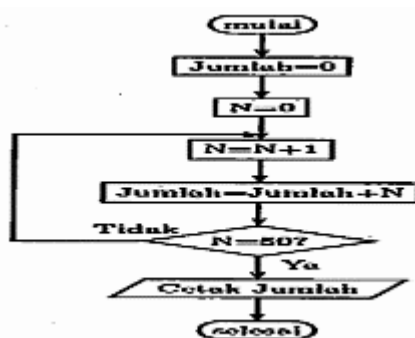
c. Detail Diagrams

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

D. Diagram alir program (flowchart)

1. Pengertian

Menurut (Suyanto, 2018) “Flowchart adalah representasi secara diagram yang menggambarkan urutan operasi dalam menyelesaikan suatu masalah”. Flowchart biasanya dibuat pada tahap awal sebelum pembuatan program. Flowchart juga merupakan fasilitas komunikasi antara programmer dan pebisnis. Selain itu flowchart juga berguna untuk memahami logika program apalagi yang panjang dan kompleks. Dengan adanya flowchart program dibuat dengan mudah.



Sumber : Suyanto, 2018

Gambar II.5 Flowchart penjualan 50 bilangan asli pertama

Keuntungan yang diperoleh dengan pemanfaatan *flowchart* adalah

- a. Komunikasi : *Flowchart* dapat menjadi sarana komunikasi bagi pihak-pihak yang terlibat dalam pengembangan program.
- b. Analisis : *Flowchart* dapat digunakan untuk analisis masalah.
- c. Penulisan program : *Flowchart* membantu konsentrasi penulisan program karena alurnya sudah jelas.
- d. Pelacakan kesalahan : *Flowchart* membantu juga dalam proses pelacakan kesalahan program.
- e. Pemeliharaan program : Bagian program yang diubah dapat ditentukan dengan melihat *flowchart* sehingga lebih efisien waktu.

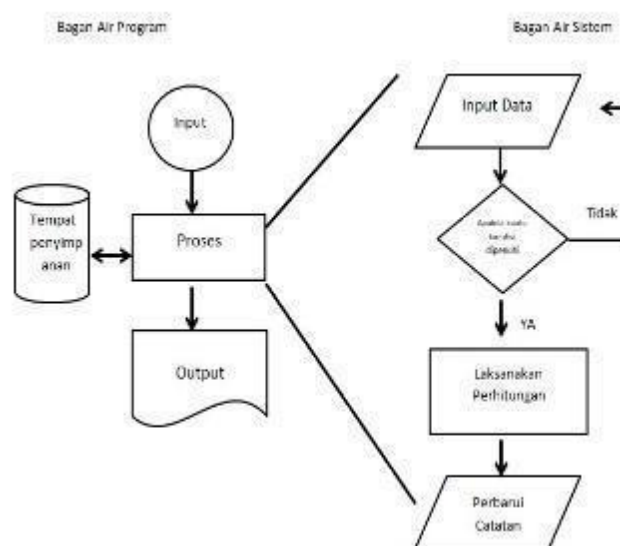
Keterbatasan *Flowchart*

- a. Logika rumit : Kadang logika program menjadi sedemikian kompleks sehingga pengguna *flowchart* menjadi begitu rumit.
- b. Modifikasi : Jika *flowchart* perlu dimodifikasi maka harus dilakukan penggambaran yang seutuhnya.
- c. Reproduksi : Karena *flowchart* berbentuk simbol-simbol dan tidak dapat diketik saja, maka proses produksi dalam media lain menjadi masalah tersendiri.
- d. Inti dari 'apa yang dikerjakan' dengan mudah menjadi kabur karena terjadi perincian 'bagaimana mengerjakannya'.

2. Bentuk *flowchart*

a. Program *Flowchart*

Menurut (Dawan, 2019) *Flowchart* program adalah baganalir yang menjelaskan secara rinci langkah-langkah dari proses program. Program *flowchart* dibuat dari deriviksi bagan alir sistem. Bagan alir logika program digunakan untuk menjelaskan tiap-tiap langkah di dalam program komputer secara logika. Bagian alir logika program dipersiapkam oleh analisis sistem.



Sumber : Dawan, 2019

Gambar II. 6 Bagan alir logika program

b. Sistem *Flowchart*

Menurut (Dawan, 2019) menjelaskan bahwa “Flowchart sistem atau bagan alir sistem merupakan bagan yang menggambarkan arus pekerjaan secara keseluruhan dari sistem”. Bagian ini menjelaskan setiap urutan dari prosedur-prosedur yang ada di dalam suatu sistem. Flowchart sistem menunjukkan apa yang dikerjakan di sistem.

Contoh : *flowchart* sistem dalam suatu pabrik, alur kerja produksi suatu barang, dan sebagainya.

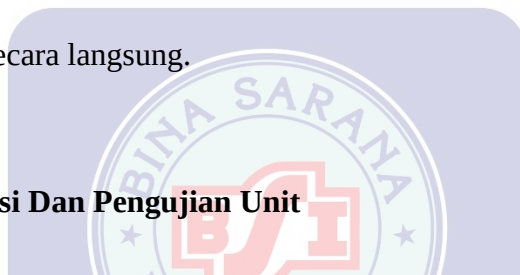
3. Teknik pembuatan

a. *General Way*

General way adalah teknik pembuatan *flowchart* dengan menggunakan proses pengulangan secara tidak langsung (*non direct loop*).

b. *Iteration Way*

Iteration Way adalah teknik pembuatan *flowchart* dengan menggunakan proses secara langsung.



E. Implementasi Dan Pengujian Unit

Testing atau pengujian unit menurut (Iriadi & Rosdiana, 2017) yaitu “Pengujian perangkat lunak yaitu dengan menggunakan metode *black box testing*. Metode uji coba *blackbox testing* memfokuskan keperluan fungsional dari *software* Uji coba *blackbox testing* berusaha menemukan kesalahan dalam fungsi-fungsi yang salah atau hilang”. *Black box* itu sendiri adalah menurut Mustaqbal, Firdaus, & Rahmadi, 2015 “pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, pengujian dapat mengartikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program.”