

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

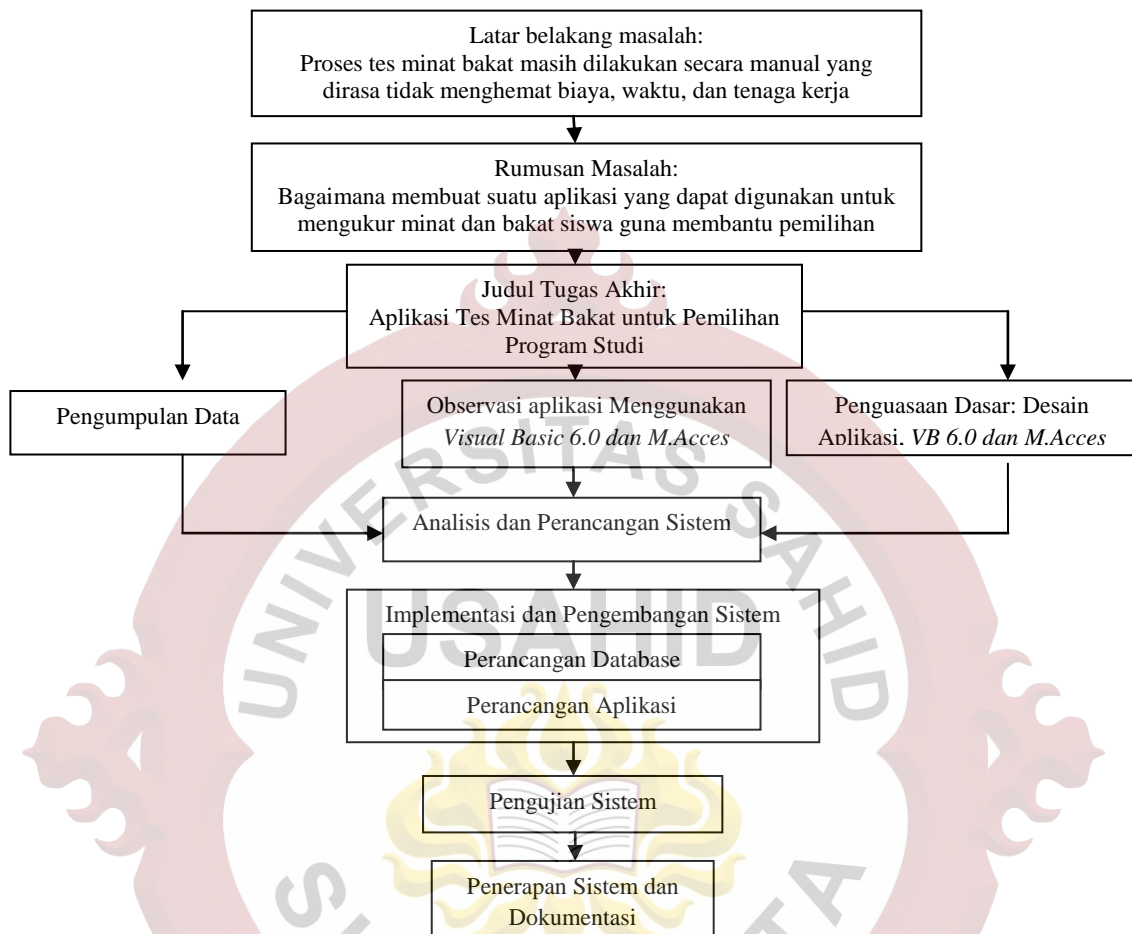
Adapun artikel atau tulisan dan kegiatan yang berhubungan dengan Tugas Akhir ini adalah Dina Andayati (2012). Dari Tugas Akhir ini penulis termotivasi untuk mengembangkan aplikasi pencarian minat dan bakat seseorang. Tugas akhir ini membahas tentang Sistem Pakar dalam Bidang Psikologi untuk Tes Minat Bakat menggunakan *Rothwell Miller Inventory Blank* (RMIB), yang merupakan instrumen tes baku/formal yang dibuat oleh Rothwell-Miller, serta telah banyak dipakai untuk mengukur bakat dan minat seseorang. Menurut hasil penelitian sistem pakar dari tes psikologi untuk kepribadian seseorang dan minat bakat, dapat menganalisa dan menarik kesimpulan, seperti halnya para pakar atau ahli. Sistem pakar yang ada dapat membantu menganalisa kepribadian seseorang berdasarkan tes dan memprediksi pekerjaan yang cocok sesuai minat dan bakat berdasarkan tes.

Namun, sistem aplikasi yang dibuat hanya menggunakan alat tes RM Blank, dimana alat tes ini hanya dapat mendeteksi minat seseorang terhadap pekerjaan. Untuk mengetahui bakat seseorang tidak dapat hanya menggunakan alat tes RM Blank. Ada beberapa alat tes yang dapat digunakan untuk mengukur bakat, salah satunya yaitu alat tes DAT. Oleh karena itu penulis mengembangkan aplikasi dengan menambahkan alat tes DAT sebagai pengukur bakat, dan RM Blank sebagai pengukur minat. Kedua alat tes tersebut dipadukan menjadi satu, sehingga diketahui minat seseorang apakah sesuai dengan bakat yang dimiliki.

#### **2.2 Kerangka Pemikiran**

Kerangka pemikiran merupakan narasi (uraian) / pernyataan tentang kerangka konsep pemecahan masalah yang telah diidentifikasi atau dirumuskan. Kerangka pemikiran dibuat berdasarkan pertanyaan penelitian dan merepresentasikan suatu himpunan dari beberapa konsep serta hubungan diantara konsep-konsep tersebut.

Kerangka pemikiran Aplikasi Tes Minat Bakat untuk Pemilihan Program Studi terlihat pada Gambar 2.1.



Gambar 2.1 Diagram Kerangka Pemikiran

## 2.3 Landasan Teori

### 2.3.1 Pengertian Minat

Minat adalah kecenderungan yang menetap untuk memperhatikan dan mengenang beberapa aktivitas. Seseorang yang berminat terhadap aktivitas akan memperhatikan aktivitas itu secara konsisten dengan rasa senang Syaiful Bahri (2008:132). Sementara itu, Slameto (2010:180), berpendapat bahwa minat adalah suatu rasa lebih suka dan ketertarikan pada suatu hal atau aktivitas tanpa ada yang menyuruh.

Umumnya minat seseorang terhadap sesuatu akan diekspresikan melalui kegiatan atau aktivitas yang berkaitan dengan minatnya. Indikator minat dapat dilihat dengan cara menganalisa kegiatan-kegiatan yang dilakukan individu yang disenanginya, karena minat merupakan motif yang dipelajari yang mendorong individu untuk aktif dalam kegiatan tertentu. Agus Sujanto (2004:92) mengatakan minat sebagai sesuatu pemusatan perhatian yang tidak sengaja yang terlahir dengan penuh kemauan dan tergantung dari bakat dan lingkungan. Pernyataan-pernyataan tersebut mengindikasikan bahwa minat dicirikan dengan adanya pemusatan perhatian atau meningkatnya perhatian terhadap sesuatu.

Berdasarkan beberapa pendapat dari para ahli di atas, minat dapat diartikan sebagai kecenderungan sifat yang terorganisir berdasarkan dari pengalaman seseorang, yang mendorong seseorang atau individu untuk mencari keterangan atau fakta-fakta dari sebuah objek, aktivitas atau kegiatan, pemahaman, *skill*, tujuan perhatian atau murni ingin mahir dalam hal tertentu.

### **2.3.2 Pengertian Bakat**

Istilah bakat dalam bahasa Inggris *aptitude*, disebut juga *talent*. Suatu bakat adalah suatu konsistensi karakteristik yang menunjukkan kapasitas seseorang untuk menguasai suatu pengetahuan khusus (dengan latihan), ketrampilan atau serangkaian respon yang terorganisir. Misalnya, kemampuan musikal (bakat musik), atau mengerjakan tugas-tugas mekanik Ki Fudaryatanta,(2010:01).

Bakat dapat diartikan sebagai kemampuan bawaan yang berupa potensi (*pottential ability*) yang masih perlu dikembangkan atau dilatih. Bakat merupakan potensi bawaan yang masih membutuhkan latihan agar dapat terwujud secara nyata. Bakat merupakan potensi terpendam dalam diri seseorang. Agar bakat dapat muncul perlu digali, ditemukan, dilatih, dan dikembangkan. Bakat memungkinkan seseorang untuk mencapai prestasi dalam bidang tertentu, akan tetapi harus ditunjang dengan minat, latihan, pengertian, pengetahuan, pengalaman, dan dorongan.

### 2.3.3 Pengertian Tes Minat Bakat

Tes minat bakat adalah tes khusus yang dirancang untuk mengetahui potensi kecerdasan umum individu, kemampuan khusus (misalnya kelebihan di bidang administratif atau kreatif), dan potensi kepribadian yang mendominasi perilaku seseorang sehingga diharapkan hasil tes dapat mengungkapkan apakah seseorang berada di tempat yang sesuai dengan karakter dan potensi serta minat orang tersebut Dewa Ketut (2003,6).

### 2.3.4 Alat Tes *Differential Aptitude Test* (DAT)

Seri tes multiple bakat DAT yang dalam bahasa Indonesia dapat dipakai istilah tes perbedaan bakat merupakan salah satu seri tes *multiple* bakat yang paling banyak dipakai dalam bidang pendidikan dan kerja. DAT adalah alat tes yang digunakan untuk mengukur bakat seseorang secara spesifik. Alat tes DAT disusun oleh George K. Bennet dan Harold G. Wesman dari USA.

Macam- macam subtes DAT adalah sebagai berikut Ki Fudyartanta (2010, 26):

- 1) Penalaran Verbal (*Verbal Reasoning*), dirancang untuk mengukur kemampuan berpikir *abstrak, generalisasi, dan konstruktif* dengan memahami konsep verbal.
- 2) Kemampuan Aritmatik (*Numerical Ability*), didesain untuk mengukur kemampuan memahami hubungan numerik dan memecahkan masalah yang berhubungan dengan masalah bilangan (*numerik*).
- 3) Penalaran Abstrak (*Abstract Reasoning*), disusun untuk mengukur penalaran nonverbal.
- 4) Kecepatan dan Keakuratan Klerikal (*Clerical Speed and Accuracy*), tes ini dirancang untuk mengukur keakuratan (ketepatan, ketelitian) dan kecepatan respon dalam tugas-tugas pekerjaan yang membutuhkan persepsi sederhana.
- 5) Penalaran Mekanik (*Mechanical Reasoning*), tes ini mengukur pemahaman prinsip-prinsip mekanik dan fisika dalam situasi yang *familier*.

- 6) Hubungan Spasial atau Ruang (*Space Relation*), dirancang untuk mengukur *visualisasi* terhadap kontruksi objek 3 dimensi yang dibangun pada pola dua dimensi.

### 2.3.5 Alat Tes *The Rothwell Miller Blank* (RM *Blank*)

*RM Blank* merupakan alat tes yang digunakan untuk mengetahui minat seseorang terhadap jenis pekerjaan. Disusun oleh Rothwell pada tahun 1947 yang terbagi dalam 9 jenis pekerjaan, dan Tahun 1958 RM Blank diperbaiki oleh Kenneth Miller sehingga menjadi 12 kategori pekerjaan. Tujuan dari tes ini adalah untuk mengukur minat seseorang berdasar sikap seseorang terhadap pekerjaan yang bersangkutan.

## 2.4 Basis Data

Beberapa literatur, basis data telah didefinisikan dengan cara yang berbeda-beda. Salah satu definisi cukup lengkap dan baik tentang istilah basis data adalah pengertian yang diberikan oleh James Martin (1975) yang dikutip oleh Abdul Kadir (2002:254) yaitu sebagai berikut:

*“A database may be defined as a collector of interrelated data stored together without harmful or unnecessary redundancy to serve one or more applications in an optimal fashion; the data are stored so that they are independent of programs with use the data; a common and controlled approach its used in adding new data in modifying and retrieving existing data within the database”.*

Jika ingin memahami pengertian di atas, maka istilah basis data dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu satu kerangkapan data (kalaupun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol (*controlled redundancy*)), data disimpan dengan cara-cara tertentu sehingga mudah untuk digunakan atau ditampilkan kembali, data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya, data disimpan sedemikian rupa sehingga



proses penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

Berdasarkan pengertian tersebut, maka suatu basis data mempunyai beberapa kriteria penting yang harus dipenuhi, yaitu:

- a) Berorientasi pada data (*data oriented*) dan bukan berorientasi pada program (*program oriented*) yang akan digunakannya. Basis data harus disimpan secara terpisah dengan program aplikasinya dan pada umumnya, paket-paket aplikasi pengelolaan basis data (*Database Management System / DBMS*) yang tersedia telah dirancang sedemikian rupa sehingga basis data akan disimpan sebagai sekumpulan *file* terpisah dengan program yang mengaksesnya.
- b) Data dapat digunakan oleh pemakai yang berbeda-beda atau beberapa program aplikasi tanpa perlu mengubah basis data. Data-data yang tersimpan dalam basis data akan digunakan oleh pemakai, yaitu individu-individu yang berbeda, sesuai dengan area kerjanya masing-masing. Sebagai contoh, pada suatu universitas, pemakai dapat terdiri atas karyawan yang bekerja di bagian akademik, karyawan dibagian keuangan, perpustakaan, kepegawaian, kemahasiswaan, dan lainnya.
- c) Data dalam basis data dapat berkembang dengan mudah baik volume maupun strukturnya. Data-data dalam basis data akan mengalami perkembangan dari waktu ke waktu. Sebuah Universitas misalnya, data-data mahasiswa akan selalu bertambah setiap awal tahun akademik karena penambahan mahasiswa baru (juga mengalami pengurangan karena ada mahasiswa yang lulus), data pengambilan mata kuliah akan bertambah setiap awal semester, data perolehan nilai mahasiswa akan bertambah setiap akhir semester, dan lainnya.
- d) Struktur basis data juga dapat mengalami perubahan seiring dengan kebutuhan subsistem-subsistem pengolahan data yang baru. Misalnya ketika diinginkan untuk mengembangkan subsistem pengolahan data praktikum di laboratorium, maka data-data yang baru bisa ditambahkan

tanpa merubah struktur basis data yang telah ada dengan tetap menjaga hubungan antar datanya.

- e) Data yang ada dapat memenuhi kebutuhan sistem-sistem baru secara mudah. Ketika terjadi penambahan/perubahan kebutuhan sistem yang baru, maka data-data dalam basis data harus dapat memenuhinya. Data-data yang telah tersimpan sebagai basis data harus tetap dapat digunakan tanpa perlu mengubahnya. Hal ini dapat terjadi hanya jika basis data dirancang sedemikian rupa sehingga ketika muncul kebutuhan-kebutuhan baru, data yang telah tersimpan tetap dapat digunakan tanpa harus dirubah dan data-data baru dapat ditambahkan dengan tetap saling berhubungan.
- f) Data dapat digunakan dengan cara yang berbeda-beda. Akses terhadap data-data dalam basis data dapat dilakukan dengan cara-cara yang berbeda. Data dalam basis data dapat diakses menggunakan program aplikasi, menggunakan instruksi-instruksi yang bersifat interaktif, menggunakan bahasa *query*, dan lainnya.
- g) Kerangkapan data (*data redundancy*) minimal. Kerangkapan data merupakan permasalahan kritis dalam basis data. Data-data dalam basis data semestinya tidak perlu disimpan secara berulang. Kerangkapan data akan mengakibatkan permasalahan yang menyulitkan ketika dilakukan pengolahan data di kemudian hari.

Keenam kriteria tersebut telah membedakan secara nyata antara pengorganisasian data secara basis data (*database processing*) dan pengelolaan data dalam *file* tradisional (*file processing*), yaitu:

- a) Hanya dapat digunakan oleh satu program aplikasi.
- b) Berhubungan dengan suatu persoalan tertentu untuk sistem yang direncanakan.
- c) Perkembangan data hanya mungkin terjadi pada volume data saja.
- d) Hanya dapat digunakan dengan satu cara tertentu saja.
- e) Kerangkapan data terlalu sering muncul.

### 2.4.1 Normalisasi

Menurut Edhy Sutanta (2004:54-55), istilah normalisasi berasal dari E.F Codd, salah seorang perintis teknologi basis data. Selain dipakai sebagai metodologi tersendiri untuk menciptakan struktur tabel (relasi) dalam basis data. Normalisasi memberikan panduan yang sangat membantu bagi pengembang untuk mencegah terciptanya struktur tabel yang kurang fleksibel atau mengurangi keefisienan

Normalisasi *database* biasanya jarang dilakukan dalam *database* skala kecil dan dianggap tidak diperlukan pada penggunaan personal. Namun seiring dengan berkembangnya informasi yang dikandung dalam sebuah *database*, proses normalisasi akan sangat membantu dalam menghemat ruang yang digunakan oleh setiap tabel didalamnya, sekaligus mempercepat proses permintaan data. Berikut ini dipaparkan metodologi logis sederhana untuk menormalkan model data dalam sebuah *database*, diiringi contoh pembuatan *database* untuk tugas-tugas mata kuliah dalam sebuah fakultas dengan atribut yang disederhanakan. Proses normalisasi model data dapat diringkas sebagai berikut:

- a) Menemukan entitas-entitas utama dalam model data.
- b) Menemukan hubungan antara setiap entitas.
- c) Menentukan atribut yang dimiliki oleh masing-masing entitas.

Normalisasi model data dilakukan dengan mengikuti langkah-langkah sederhana, mengubahnya agar memenuhi apa yang disebut sebagai bentuk normal pertama, kedua, lalu ketiga secara berurutan.

### 2.4.2 Teknik Normalisasi

Sebuah model data dikatakan memenuhi bentuk normal apabila setiap atribut yang dimilikinya memiliki satu dan hanya satu nilai, apabila ada atribut yang memiliki nilai lebih dari satu, atribut tersebut adalah kandidat untuk menjadi entitas tersendiri dan biasanya disebut bentuk tidak normal. Bentuk normalisasi meliputi:

#### 1. Bentuk Normal Pertama (1NF)

Sebuah model data dikatakan memenuhi bentuk normal pertama apabila setiap atribut yang dimilikinya memiliki satu dan hanya satu nilai. Apabila ada



atribut yang memiliki nilai lebih dari satu, atribut tersebut adalah kandidat untuk menjadi entitas tersendiri.

Entitas utama untuk *database* stok barang tentu saja kode barang. Atribut kode barang mencantumkan harga beli di mana harga beli tersebut diurutkan berdasarkan kode barang. Apabila input harga beli untuk sebuah kode barang tidak sama dengan harga beli yang ada, maka secara otomatis harga beli tersebut akan dijumlahkan dengan harga beli yang telah ada kemudian dibagi dua.

## 2. Bentuk Normal Kedua (2NF)

Sebuah model data dikatakan memenuhi bentuk normal kedua apabila ia memenuhi bentuk normal pertama dan setiap atribut *non-identifier* sebuah entitas bergantung sepenuhnya hanya pada semua *identifier* entitas tersebut. Misalnya sebuah kode barang akan tetap ada meskipun stok barang sudah tidak ada lagi atau nol. Dalam hal ini, kode barang adalah entitas tersendiri.

## 3. Bentuk Normal Ketiga (3NF)

Sebuah model data dikatakan memenuhi bentuk normal ketiga apabila ia memenuhi bentuk normal kedua dan tidak ada satupun atribut *non-identifying* (bukan pengidentifikasi unik) yang bergantung pada atribut *non-identifying* lain. Apabila ada, dipisahkan salah satu atribut tersebut menjadi entitas baru, dan atribut yang bergantung padanya menjadi atribut entitas baru tersebut.

Sebagaimana dalam dua langkah normalisasi sebelumnya, jenis kebergantungan seperti ini dapat dihilangkan dengan membuat entitas baru lagi (yang tidak akan diciptakan karena tiga entitas sudah cukup banyak untuk satu artikel).

Model terakhir yang penulis dapat ini telah memenuhi bentuk normal ketiga (*third normal form*) dan siap dikonversi menjadi tabel. Namun sebelumnya, kita perlu membahas berbagai jenis relasi yang kerap ditemui dalam pemodelan data, termasuk yang kita temui dalam contoh model data kali ini.

### 2.5 Flowchart

*Flowchart* merupakan teknik analitis yang digunakan untuk menjelaskan aspek-aspek sistem informasi secara jelas, tepat dan logis. *flowchart* menggunakan serangkaian simbol setandar untuk menguraikan prosedur

pengolahan transaksi yang digunakan oleh sebuah perusahaan, sekaligus menguraikan aliran data dalam sebuah sistem.

*Flowchart* digunakan sebagai alat bantu komunikasi, maka dalam membuat sistem *flowchart* ini harus mampu menunjukkan arus pekerjaan secara keseluruhan dari yang akan dibuat. Pembuatan model *flowchart* yang berbeda sesuai dengan perspektif pemakai sehingga dikenal ada 4 jenis diagram alir secara umum:

- a) Diagram Alir Dokumen, menunjukkan kontrol dari sebuah sistem aliran dokumen.
- b) Diagram Alir Data, menunjukkan kontrol dari sebuah sistem aliran data.
- c) Diagram Alir Sistem, menunjukkan kontrol dari sebuah sistem aliran secara fisik.
- d) Diagram Alir Program, menunjukkan kontrol dari sebuah program dalam sebuah sistem.

#### **2.5.1 *Flowchart* Sistem**

*Flowchart* Sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. *flowchart* merupakan deskripsi secara grafik dari urutan prosedur-prosedur yang terkomposisi yang membentuk suatu sistem.

*Flowchart* Sistem terdiri dari data yang mengalir melalui sistem dan proses yang mentransformasikan data. Data dan proses dalam *flowchart* sistem dapat digambarkan secara *online* atau *offline* (misalnya mesin ketik, *cash register* atau kalkulator). Contoh sederhana untuk *flowchart* sistem dapat dilihat pada Tabel 2.1.

#### **2.5.2 *Flowchart* Program**

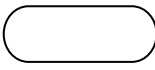
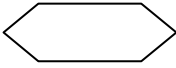
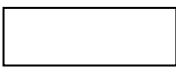





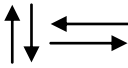
*Flowchart* Program dihasilkan dari *Flowchart* Sistem. *Flowchart* Program merupakan keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur sesungguhnya dilaksanakan. *Flowchart* ini menunjukkan setiap langkah program atau prosedur dalam urutan yang tepat saat terjadi. *Programmer* menggunakan *flowchart* program untuk menggambarkan urutan instruksi dari

program komputer. Analisis Sistem menggunakan *flowchart* program untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi. Suatu contoh *flowchart* program dapat dilihat pada Tabel 2.2.

Tabel 2.1 Simbol-Simbol *Flowchart* Sistem

SIMBOL	KETERANGAN
	<b>Input/Output.</b> Simbol <i>Input/Output</i> digunakan untuk menyatakan dan mewakili data masukan atau keluaran.
	<b>Manual Input.</b> Simbol <i>Manual Input</i> digunakan untuk menginputkan data secara manual (Contoh: memasukkan data melalui <i>keyboard</i> komputer)
	<b>Process.</b> Simbol <i>Process</i> atau Pengolahan digunakan untuk mewakili suatu proses, seperti pengolahan aritmatika atau pemindahan data.
	<b>Disk Magnetic.</b> Simbol <i>Disk Magnetic</i> menunjukkan bahwa data disimpan secara permanen di dalam <i>disk</i> magnetik
	<b>Online Storage.</b> Simbol <i>Online Storage</i> digunakan untuk menyimpan akses sistem secara langsung.
	<b>Flow Lines.</b> Simbol <i>flow lines</i> digunakan untuk menunjukkan aliran atau arus dari proses.
	<b>Display.</b> Simbol <i>Display</i> berisi informasi yang ditampilkan dengan monitor, terminal, atau layar.
	<b>Document.</b> Simbol <i>document</i> adalah sebagai hasil <i>output</i> /keluaran dari sistem

Tabel 2.2 Simbol-Simbol *Flowchart* Program





SIMBOL	KETERANGAN
	<b>Terminal Point Symbol.</b> Simbol ini digunakan untuk menunjukkan awal dan akhir dari suatu proses.
	<b>Preparation Symbol.</b> Simbol ini digunakan untuk memberi nilai awal Suatu besaran atau <i>variable</i> .
	<b>Process Symbol.</b> Simbol Proses atau Pengolahan digunakan untuk mewakili suatu proses, seperti pengolahan aritmatika atau pemindahan data.
	<b>Predefined Process Symbol.</b> Simbol Proses Terdefinisi digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain atau untuk proses yang detailnya dijelaskan terpisah, misalnya berbentuk <i>subroutine</i> .
	<b>Decision Symbol.</b> Simbol Keputusan digunakan untuk mewakili operasi perbandingan logika atau suatu penyeleksian kondisi di dalam program.
	<b>Input/Output Symbol.</b> Simbol <i>Input/Output</i> digunakan untuk menyatakan dan mewakili data masukan atau keluaran.
	<b>Connector Symbol.</b> Simbol Penghubung digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama.
	<b>Off-page Connector.</b> Simbol Penghubung Halaman lain digunakan untuk menunjukkan hubungan arus proses yang terputus dengan sambungannya ada di halaman yang lain.
	<b>Flow Lines Symbol.</b> Simbol Garis Alir digunakan untuk menunjukkan aliran atau arus dari proses.

## 2.6 Data Flow Diagram (DFD)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD menggambarkan penyimpanan data dan proses yang mentransformasikan data. DFD menunjukkan hubungan antara data pada sistem dan proses pada sistem. Berikut ini adalah simbol-simbol dalam merancang DFD. DFD didesain untuk menunjukkan sebuah sistem yang terbagi-bagi menjadi suatu bagian sub-sistem yang lebih kecil adan untuk menggarisbawahi arus data antara kedua hal yang tersebut diatas. Diagram ini lalu dikembangkan untuk melihat lebih rinci sehingga dapat terlihat model-model yang terdapat di dalamnya.

Tabel 2.3 Simbol-Simbol DFD

Simbol	Keterangan
	Merepresentasikan sumber data ( <i>Entity</i> ).
	Merepresentasikan aliran data.
	Merepresentasikan transformasi/ proses aliran data (sistem).
	Merepresentasikan tempat untuk menyimpan data ( <i>file</i> ).

## 2.7 Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data selain digunakan untuk dokumentasi dan mengurangi redundansi, juga dapat digunakan untuk:

1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan data yang disimpan dalam *file-file*.



4. Mengembangkan logika untuk proses-proses diagram aliran data.

Kamus Data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Tahap analisis sistem, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Tahap perancangan sistem, kamus data digunakan untuk merancang *input*, merancang laporan-laporan dan *database*. Kamus data dibuat berdasarkan arus data yang ada di DFD. Kamus Data mendefinisikan elemen data dengan fungsi sebagai berikut:

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran (misalnya alamat diuraikan menjadi Kode Pos, Kelurahan, Kecamatan, Kabupaten/Kota, dan Provinsi)
3. Mendeskripsikan komposisi penyimpanan data.
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran.
5. Mendeskripsikan hubungan detail antar penyimpanan (yang akan menjadi titik perhatian dalam *entity-relationship diagram*).

#### 2.7.1 Elemen-Elemen Data

Kamus data harus dapat mencerminkan keterangan yang jelas tentang data yang akan dicatat dan untuk maksud keperluan ini, maka kamus data harus memuat hal-hal berikut:

1. **Nama arus data**, karena kamus data dibuat berdasarkan arus data yang mengalir di DFD, maka nama dari arus data juga harus dicatat di kamus data.
2. **Alias**, alias atau nama lain dari data dapat dituliskan bila nama lain ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya. Misalnya bagian pembuat faktur dan langganan menyebut bukti penjualan sebagai faktur, sedangkan bagian gudang menyebutnya sebagai tembusan permintaan

persediaan. Baik faktur dan tembusan permintaan persediaan ini mempunyai struktur data yang sama, tetapi mempunyai struktur yang berbeda.

3. **Bentuk data**, telah diketahui bahwa arus data dapat mengalir:
  - a) Dari kesatuan luar ke suatu proses, data yang mengalir ini biasanya tercatat di suatu dokumen atau formulir.
  - b) Hasil dari suatu proses ke kesatuan luar, data yang mengalir ini biasanya terdapat di media laporan atau *query* tampilan layar atau dokumen hasil cetakan komputer.
  - c) Hasil suatu proses ke proses yang lain, data yang mengalir ini biasanya dalam bentuk variabel atau parameter yang dibutuhkan oleh proses penerimanya.
  - d) Hasil suatu proses yang direkamkan ke simpanan data, data yang mengalir ini biasanya berbentuk suatu variabel.
  - e) Dari simpanan data dibaca oleh suatu proses, data yang mengalir ini biasanya berupa suatu *field*.

Dengan demikian bentuk dari data yang mengalir dapat berupa: dokumen dasar atau formulir, dokumen hasil cetakan komputer, laporan tercetak, tampilan di layar monitor, variabel, parameter, *field*.

4. **Arus data**, arus data menunjukkan dari mana data mengalir dan ke mana data akan menuju. Keterangan ini perlu dicatat di kamus data agar mudah mencari arus data di DFD.
5. **Penjelasan**, digunakan untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di kamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut. Misalnya nama dari arus data adalah Tembusan Permintaan Persediaan, maka dapat lebih dijelaskan sebagai tembusan dari faktur penjualan untuk meminta barang dari gudang.
6. **Periode**, periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di kamus data karena dapat digunakan untuk mengidentifikasi kapan *input* data harus dimasukkan ke sistem, kapan

proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

7. **Volume**, volume yang perlu dicatat di kamus data adalah tentang volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya rata-rata arus data yang mengalir dalam satu periode tertentu dan volume puncak menunjukkan volume yang terbanyak. Volume ini digunakan untuk mengidentifikasi besarnya simpanan luar yang akan digunakan, kapasitas dan jumlah dari alat *input*, alat pemroses dan alat *output*.
8. **Struktur data**, struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari item-item apa saja yang terdapat di dalam data.

## 2.8 *Microsoft Visual Basic 6.0*

### 2.8.1 *Pengenalan Visual Basic 6.0*

Menurut Jones Phil (2001) yang dikutip oleh Bunafit Nugroho (2007:9), *visual Basic* adalah bahasa Program yang berbeda dengan bahasa program basic konvensional yang telah kita kenal umumnya. Pemrograman *Visual Basic* untuk merancang suatu aplikasi *project*, terlebih dahulu harus memperkirakan format *output* yang diperlukan, selanjutnya diikuti dengan penulisan kode-kode program sesuai dengan rancangan *output* yang dimiliki. Konsep tersebut dikenal dengan konsep *Top-Down*, artinya alur merancang suatu aplikasi program dilakukan dengan membuat *outputnya* terlebih dahulu kemudian menuliskan kode program. Segala kegiatan merancang bentuk *output* serta penulisan kode program dan proses program dapat diikuti dengan mudah dan jelas sehingga dikenal dengan istilah pemrograman visual.

Merancang format *output* di *Visual Basic* berbeda dengan bahasa *basic* biasa yang konvensional, karena di *Visual Basic* ini membentuk *output* dengan sekumpulan *object*, *object-object* tersebut telah disediakan sebagai fasilitas *tool* di *Visual Basic*, kumpulan *object* tersebut dinamakan *ToolBox*. Setiap *object* yang ada pada *toolbox* mempunyai karakter dan perilaku khusus, karakter perilaku khusus yang ada kemudian diistilahkan dengan *Control*. *Control* ini bisa diatur

dengan menggunakan bantuan *Windows Property*. Selanjutnya *object* dengan *control* tersebut disusun, didesain pada media yang dinamakan *Form*.

### 2.8.2 *Integrated Development Integration (IDE)*

IDE adalah tempat bekerja untuk menghasilkan program aplikasi. IDE memungkinkan penulis dapat menggunakan banyak *tool* yang dapat diambil dengan mudah. Layar ini adalah lingkungan pengembangan aplikasi *Visual Basic* yang nantinya akan digunakan untuk membuat program-program aplikasi dengan *Visual Basic*. Layar *Visual Basic* hampir sama dengan layar program-program aplikasi *windows* pada umumnya, terutama jika pernah menggunakan bahasa pemrograman visual lainnya, seperti *Microsoft Visual Foxpro*, *Microsoft Access*, dan lain sebagainya.

#### a) ***Menu Bar***

Menu ini adalah menu utama yang menampilkan perintah-perintah yang terdapat pada *visual basic*. *Menu bar* ini terdapat *file menu* yang khas untuk *software under windows* seperti *file*, *edit*, *program*, dan seterusnya. Setiap *title* mempunyai *item menu* masing-masing yang dikelompokkan berdasarkan *title menu* yang ada, di atas *menu bar*.

#### b) ***ToolBar***

*Toolbar* yaitu sekumpulan tombol yang berfungsi sebagai tombol cepat (*Shortcut*) untuk menjalankan perintah dan mengendalikan lingkungan pemrograman *Visual Basic*. Tombol-tombol ini merupakan perintah-perintah yang sering digunakan dan terdapat pula pada menu *Visual Basic*.

#### c) ***ToolBox***

*Toolbox* merupakan kumpulan *object* yang digunakan untuk merancang sebuah *output* program. Karena masing-masing *object* mempunyai sifat yang khas, dan dengan sifat yang khas tersebut penulis tinggal memberikan kontrolnya, sehingga menjadi suatu aplikasi program yang penulis inginkan, maka *object* ini disebut dengan *Control*.



d) **Form**

*Form* adalah bahan pembuatan *windows* dan meletakkan *control* pada *form*. *Control* ini misalnya tombol *check box*, *button*, label, dan sebagainya, pada *form* ini penulis dapat membuat aplikasi yang diinginkan.

### 2.8.3 Keistimewaan *Visual Basic 6.0*

Menurut Bunafit Nugroho (2007:11), keistimewaan *Visual Basic 6.0* adalah:

- a) Menggunakan *platform* pembuatan program yang diberi nama *Developer Studio*, yang memiliki tampilan dan sarana yang sama dengan *Visual C++*.
- b) Memiliki *compiler* yang andal yang dapat menghasilkan *file executable* yang lebih cepat dan lebih efisien dari sebelumnya.
- c) Memiliki tambahan beberapa sarana *Wizard* yang baru. *Wizard* adalah sarana yang mempermudah didalam membuat aplikasi dengan mengotomatisasi tugas-tugas tertentu.
- d) Tambahan kontrol-kontrol baru yang lebih canggih serta peningkatan struktur bahasa *Visual Basic*.
- e) Kemampuan membuat *ActiveX* dan fasilitas *internet* yang lebih banyak.
- f) Sarana akses data yang lebih cepat dan andal untuk membuat aplikasi *database* yang berkemampuan tinggi.

## 2.9 *Microsoft Access*

*Microsoft Access* adalah satu program pengelolaan *database* berdaya guna dan fleksibel dijual saat ini. *Microsoft Access* dapat menangani pekerjaan di mana pun membutuhkan *record* sederhana seperti sistem manajemen bisnis yang lengkap atau data-data bagi kepegawaian.

*Microsoft Access* memungkinkan untuk mengumpulkan data, dan menyimpan dan mengatur informasi seperti halnya membuat laporan mengarah kepada kesimpulan akhir. Berikut beberapa hal yang dapat dilakukan dengan *Microsoft Access*:

1. Mengetik data langsung ke *database* atau mengimpornya dari program lain
2. Secara cepat membuat laporan semuan atau sebagian dari data
3. Mengurutkan, mengideks dan mengatur data dengan cara yang diinginkan.



## 2.10 Metode McCall

Menurut McCall yang dikutip oleh Winarsih (2009:2) mengusulkan kategori yang berguna mengenai faktor-faktor yang mempengaruhi perangkat lunak. Fokusnya terdapat pada tiga hal penting produk perangkat lunak karakteristik operasional, kemampuannya untuk beradaptasi dengan lingkungan yang baru. Faktor-faktor kualitas perangkat lunak McCall terdiri dari:

### 1. Kebenaran

Kebenaran adalah tingkat dimana program memenuhi spesifikasinya dan memenuhi sasaran.

### 2. Reliabilitas

Reliabilitas adalah tingkat dimana sebuah program dapat diharapkan melakukan fungsi yang diharapkan dengan ketelitian yang diminta.

### 3. Efisiensi

Efisiensi adalah jumlah sumber daya penghitungan kode yang diperlukan oleh program untuk melakukan fungsinya.

### 4. Integritas

Integritas adalah tingkat dimana akses ke perangkat lunak atau data oleh orang yang tidak berhak dapat di kontrol.

### 5. Usabilitas

Usabilitas adalah kemudahan mengoperasikan, menyiapkan *input*, dan menginterpretasikan *output* suatu program.

### 6. Maintainabilitas

*Maintainabilitas* adalah usaha yang diperlukan untuk mencari dan membetulkan kesalahan pada sebuah program.

### 7. Fleksibilitas

*Fleksibilitas* adalah usaha yang diperlukan untuk memodifikasi program operasional.

### 8. Testabilitas

*Testabilitas* adalah usaha yang diperlukan untuk menguji sebuah program untuk memastikan apakah program melakukan fungsi–fungsi yang dimaksudkan.

## 9. Portabilitas

*Portabilitas* adalah usaha yang diperlukan untuk memindahkan program dari satu perangkat keras dan atau lingkungan.

## 10. Reusabilitas

*Reusabilitas* adalah tingkat dimana sebuah program (bagian dari suatu program) dapat digunakan kembali di dalam aplikasi lain.

## 11. Interoperabilitas

*Interoperabilitas* adalah usaha yang diperlukan untuk merangkai satu sistem dengan yang lainnya.

Kualitas *software* diukur dengan metode penjumlahan dari keseluruhan kriteria dalam suatu faktor sesuai dengan bobot (*weight*) yang telah ditetapkan. Rumus pengukuran yang digunakan dalam metode McCall adalah:

$$F_a = w_1c_1 + w_2c_2 + \dots + w_nc_n$$

Dimana:

**F<sub>a</sub>** = nilai total dari faktor **a**

**w<sub>1</sub>** = bobot untuk kriteria ke-1

**c<sub>1</sub>** = nilai untuk kriteria ke-1

Kemudian tahapan yang harus ditempuh dalam pengukuran adalah:

**Tahap 1:** Tentukan kriteria yang digunakan untuk mengukur suatu faktor.

**Tahap 2:** Tentukan bobot (*w*) dari setiap kriteria (biasanya  $0 \leq w \leq 1$ ).

**Tahap 3:** Tentukan skala dari nilai kriteria (misalnya,  $0 \leq \text{nilai kriteria} \leq 10$ ).

**Tahap 4:** Berikan nilai pada tiap kriteria.

**Tahap 5:** Hitung nilai total dengan rumus  $F_a = w_1c_1 + w_2c_2 + \dots + w_nc_n$ .