BAB 2

LANDASAN TEORI

2.1. Pengertian Mengetik

Mengetik merupakan sebuah teknik mengenai penggunaan komputer yang harus dipelajari dan dilatih. Jika mengetik dilakukan tanpa disertai dengan pengetahuan serta keterampilan, maka boleh jadi ketikan yang dihasilkan tidak akan memuaskan dan berpotensi timbul kesalahan [1].

Jika ditelusuri secara seksama, asal mula kata mengetik sebenarnya adalah *onomatope* atau tiruan bunyi dari bunyi *tuts* atau tombol yang ditekan. Bunyi tik yang dihasilkan adalah suara dari *tuts* pada mesin yang memiliki *tuts* atau tombol yang bisa ditekan. Jadi, secara *onomatope* mengetik itu adalah mengetuk atau menekan dengan ujung jari yang menghasilkan bunyi tik-tik.

2.1.1. Metode Pengetikan

Mengetik juga memiliki beberapa metode yang bisa digunakan agar mempermudah proses pengetikan. Metode-metode ini baru bisa dilakukan jika kemampuan dalam mengetik sudah terbilang mahir. Berikut ini adalah beberapa metode pengetikan yang biasa digunakan [1]:

1. Mengetik Sistem Buta

   Mengetik dengan sistem buta adalah metode pengetikan yang biasa digunakan ketika akan menyalin naskah ke dalam komputer atau mesin tik. Ketika menyalin naskah tersebut, pandangan mata hanya tertuju pada naskah,

sama sekali tidak melihat ke arah layar komputer atau *tuts*. Metode ini bisa dikatakan hanya mengandalkan perasaan.

2. Mengetik Sistem Berirama

Suara tik yang ditimbulkan ketika menekan *tuts* menjadi hal pokok dalam sistem ini. Bunyi yang dihasilkan oleh tekanan-tekanan *tuts* harus memiliki ketukan yang sama sehingga suara hentakan yang dihasilkan menghasilkan irama. Kedua metode pengetikan itu hanya dapat dilakukan jika metode mengetik sepuluh jari sudah mahir dilakukan.

3. Mengetik Sepuluh Jari

Pada metode ini, sepuluh jari sudah memiliki peranan dalam menekan *tuts* masing-masing. Metode seperti ini harus dilatih agar jari dapat menekan *tuts* tanpa ragu-ragu. Gerakan jari harus dilakukan dengan gerakan yang teratur sehingga *tuts* akan terlihat seperti tertekan secara otomatis. Berhasil atau tidaknya metode ini sangat mengandalkan perasaan. Perasaan dapat diasah dengan latihan yang rutin.

2.1.2. Pengertian Mengetik Sepuluh Jari

Dalam bahasa Inggris, mengetik sepuluh Jari disebut juga dengan istilah Touch Typing, artinya mengetik tanpa menggunakan indra penglihatan untuk mencari *tuts keyboard* yang diinginkan. Secara spesifik, pengetik atau disebut juga dengan *typist* mengetahui lokasi *tuts keyboard* lewat memori otot (*muscle memory*) [1].

Memori otot dikenal sebagai *motor learning*, artinya suatu bentuk prosedur memori yang melibatkan konsolidasitugas motorik untuk masuk ke dalam memori melalui pengulangan gerakan otot. Ketika dilakukan gerakan berulang dari waktu ke waktu, memori otot jangka panjang diberi tugas tertentu sehingga akhirnya memungkinkan untuk melakukan suatu tugas tersebut tanpa upaya sadar. Proses ini mengurangi kebutuhan untuk perhatian dan menciptakan efisiensi maksimum dalam sistem motor dan memori. Contoh memori otot yang ditemukan dalam aktivitas sehari-hari banyak yang menjadi otomatis dan meningkat secara praktik, seperti makan, mengetik pada keyboard, naik sepeda, memainkan alat musik, bermain *video game*, atau bahkan memecahkan teka-teki kubus.

Pengetik sepuluh jari menyimpan delapan jarinya di sepanjang *tuts horizontal* bagian tengah *keyboard* (*the home row*) dan menggunakan kedelepan jari tersebut untuk meraih *tuts* lainnya yang berada di barisan keyboard bagian atas dan bawah. Dua jari lagi, yaitu jari jempol, digunakan untuk menekan tombol spasi. Akan tetapi, sebagian besar pengetik menggunakan jempol tangan kanan untuk menekan tombol spasi. Orang yang mengetik dengan sepuluh jari biasa disebut *copy typist* atau *audio typist.* Terdapat dua keahlian yang harus dimiliki *copy typist*. Yang pertama adalah mengetik dengan cepat dan yang kedua adalah mengetik dengan akurat. Kecepatan mengetik sepuluh biasanya diukur dalam dua skala, yaitu CPM (*Characters Per Minute*) dan WPM (*Words Per Minute*). CPM adalah jumlah karakter benar yang diketik dalam satu menit sedangkan WPM

ialah jumlah kata, standarnya lima huruf [23], yang diketik dalam satu menit. WPM bisa didapat dari CPM dibagi lima [25].

Di dalam pengetikan, ada aspek lain yang berhubungan dengan jari, yaitu waktu rata-rata respon. Waktu rata-rata jari merespon diukur dalam satuan *millisecond* (ms) atau seper-ribu detik. Waktu tersebut dihitung dari awal pengetikan atau karakter yang diketik sampai karakter (yang bukan salah ketik) setelahnya yang diketik [23]. Untuk waktu rata-ratanya didapat dari jumlah waktu suatu karakter yang benar diketik dibagi jumlah karakter ketikan itu.

Sebagai contoh kasus, Gambar 2.1 menunjukkan deretan huruf, yaitu baris atas, yang telah diketik dengan benar dari kiri ke kanan sedangkan baris bawah merupakan deretan *millisecond* antar penekanan. Di sini akan dihitung waktu rata-rata respon jari telunjuk tangan kiri. Berdasarkan gambar, yang ditangani oleh telunjuk tangan kiri adalah huruf B. Maka,

Jumlah yang diketik dengan tepat = 3

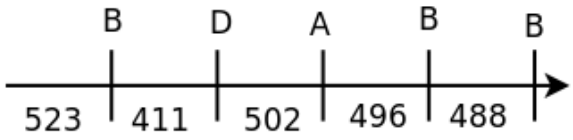
Jumlah waktu = 523 + 496 + 488 = 1507

Jadi, waktu rata-rata respon jari telunjuk tangan kiri ialah 1507 / 3 = 502 (dibulatkan). Apabila pada deret ke- tiga yang seharusnya diketik adalah B tetapi yang diketik adalah huruf A, *player* akan mendapatkan poin kesalahan dan perhitungan waktu rata-rata respon jari yang menangi huruf B adalah

Jumlah yang diketik dengan tepat = 3

Jumlah Waktu = 523 + 502 + 496 + 488 = 2009

502 dimasukkan karena setelahnya merupakan huruf yang salah diketik. Jadi, waktu rata-rata respon jari telunjuk tangan kanan adalah 2009 / 3 = 670 (dibulatkan).



Gambar 2.1. Sampel deretan huruf yang diketik.

2.2. *Game* Edukasi

Sebuah *game* merupakan sebuah lingkungan belajar interaktif yang menarik bagi pemain dengan menawarkan tantangan yang membutuhkan peningkatan tingkat penguasaan. Namun, dengan munculnya berbagai jenis permainan, kata *"game"* menjadi sulit untuk didefinisikan serta memegang berbagai denotasi dan konotasi. Masalah tersebut bisa diselesaikan secara sederhana dengan melihat sejumlah karakteristik dari *game* yang biasanya terdiri atas tujuan, kemampuan, sumber daya, sarana, interaksi, strategi, keterlibatan, pengambilan keputusan, dan kebutuhan pemecahan masalah [5].

Game edukasi adalah *software game* (*game* perangkat lunak) yang diarahkan tidak hanya untuk hiburan saja tetapi lebih mengedepankan pendidikannya. Game jenis ini mengajarkan anak-anak atau pun orang dewasa beberapa bentuk pembelajaran dengan cepat dan mudah diserap. Sementara pembelajaran di sekolah konvensional dirasa lambat dan membosankan. Salah satu contoh *game* edukasi yang dikembangkan oleh mahasiswa ialah Mbatik Yuk.

Game edukasi ini menampilkan simulasi bagaimana cara membatik. Mulai dari tahap kain polos sampai dengan kain batik yang sudah terisi dengan warna [6].

2.3. Animasi

Pengertian animasi (*animation*) pada dasarnya adalah menggerakkan objek agar tampak lebih dinamis. Sebelum era komputerisasi seperti sekarang, animasi merupakan proses yang rumit dan menyita banyak waktu dan tenaga. Film-film animasi terdahulu menggunakan ratusan sampai ribuan gambar sketsa tangan untuk membuat sebuah animasi pergerakan satu *per* satu. Tiap gambar bergerak tersebut dikenal dengan *frame*. Semakin banyak gambar yang digunakan, maka semakin halus pergerakan animasinya.

Setelah era komputer grafik seperti sekarang, proses animasi tidak lagi merupakan suatu proses yang terlalu rumit. Seorang animator 2D atau 3D cukup menganimasikan *frame* awal dan akhir dari suatu pergerakan animasi, selebihnya komputer akan mengkalkulasi gerakan di antaranya (dikenal dengan istilah *In-Between*) [15].

Gambar 2.2 menunjukkan delapan *frame* (bagian) karakter berjalan yang merupakan potongan gambar pembentuk animasi. Bila potongan gambar itu dimunculkan satu per satu, maka karakter akan terlihat berjalan seolah-olah hidup. Gambar tersebut merupakan tipe *loop* di mana *frame* ke-delapan bersambung ke *frame* ke-satu.

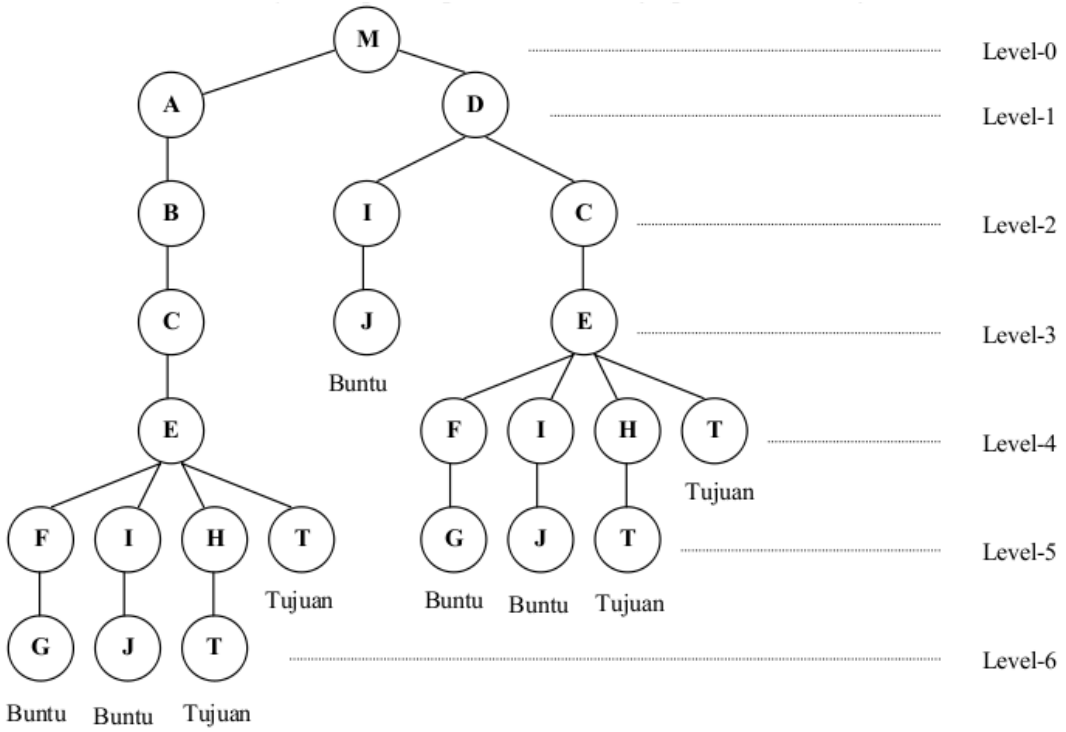


Gambar 2.2. *Frame* animasi [24].

2.4. *Depth-First Search*

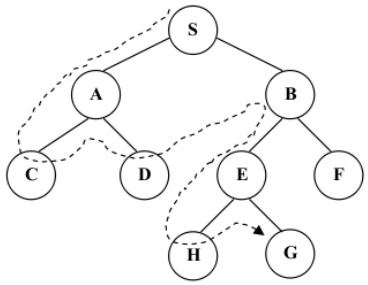
Artificial intelligence / AI (kecerdasan buatan) didefinisikan sebagai suatu mesin atau alat pintar (biasanya adalah komputer) yang dapat melakukan suatu tugas yang bilamana tugas tersebut dilakukan oleh manusia akan dibutuhkan suatu kepintaran untuk melakukannya. *Game playing* (permainan *game*) merupakan salah satu bidang AI yang sangat populer di mana mesin mempunyai intelektual untuk berpikir melawan manusia [13].

Masalah dalam suatu kasus yang melibatkan AI dapat dianggap sebagai ruang keadaan (*state space*), yaitu suatu ruang yang berisi semua keadaan yang mungkin. Salah satu cara untuk merepresentasikan ruang keadaan ialah dengan pohon pelacakan. Pohon pelacakan digambarkan secara hirarki. Simpul (*node*) yang terletak pada *level* 0 disebut simpul akar (*initial state* / *root*). Simpul akar menunjukkan keadaan awal dan memiliki beberapa percabangan yang terdiri atas beberapa simpul yg disebut simpul anak / suksesor (*child*). Simpul di *level* atas dianggap simpul *parent* oleh suksesornya. Simpul dapat berupa tujuan yang diharapkan (*goal state*) atau jalan buntu (*dead end*). Hubungan dari simpul ke simpul disebut operator. Gambar 2.3 adalah contoh dari pohon pelacakan.



Gambar 2.3. Contoh pohon pelacakan.

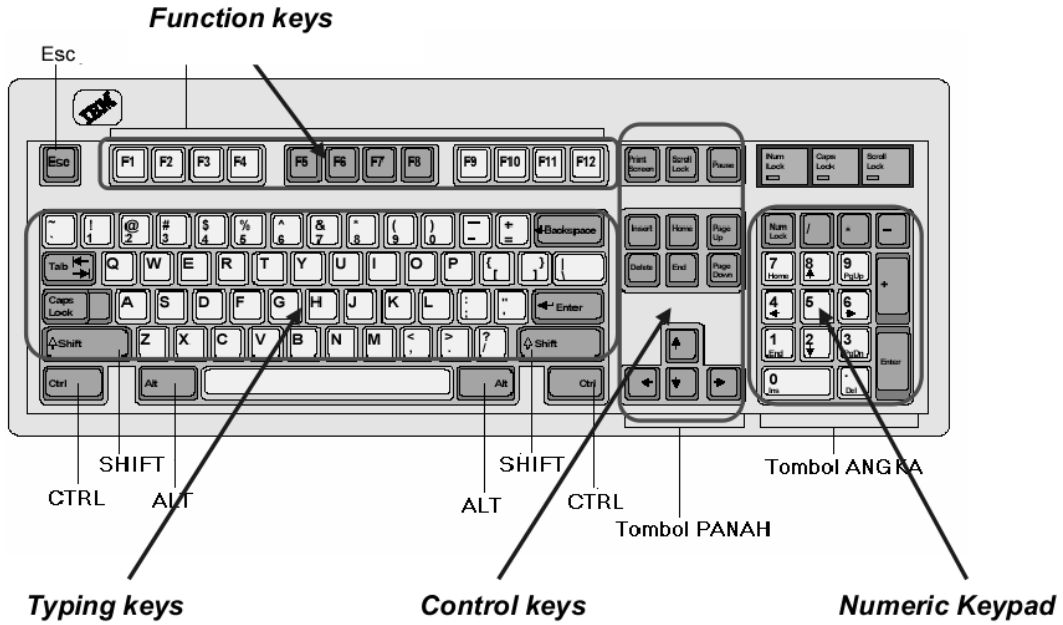
Depth-first search (DFS) merupakan metode pencarian solusi tanpa informasi (*blind search*). Pencarian dilakukan pada suatu simpul dalam setiap level dari yang paling kiri. Jika pada *level* yang terdalam solusi belum ditemukan, maka pencarian dilanjutkan pada simpul sebelah kanan. Jika pada *level* yang paling dalam tidak ditemukan solusi, maka pencarian dilajutkan pada *level* sebelumnya. Penelusuran DFS dapat dilihat pada Gambar 2.4. [13].



Gambar 2.4. Penelusuran DFS.

2.5. Pengertian *Keyboard*

Keyboard atau papan ketik komputer merupakan sebuah papan yang terdiri dari tombol-tombol, seperti huruf untuk mengetikkan kalimat, angka untuk menyatakan bilangan, serta simbol-simbol khusus yang berguna untuk membuat ekspresi tertentu. Pada *keyboard* terdapat tombol-tombol huruf A sampai Z, 0 sampai 1, dan tombol-tombol dengan simbol khusus, misalnya +, &, dan @. Contoh ilustrasi *keyboard* standar dan *keyboard* laptop dapat dilihat pada Gambar 2.5 dan 2.6 [1].



Gambar 2.5. *Keyboard* qwerty standar.



Gambar 2.6. *Keyboard* qwerty laptop.

2.5.1. Sejarah Keyboard

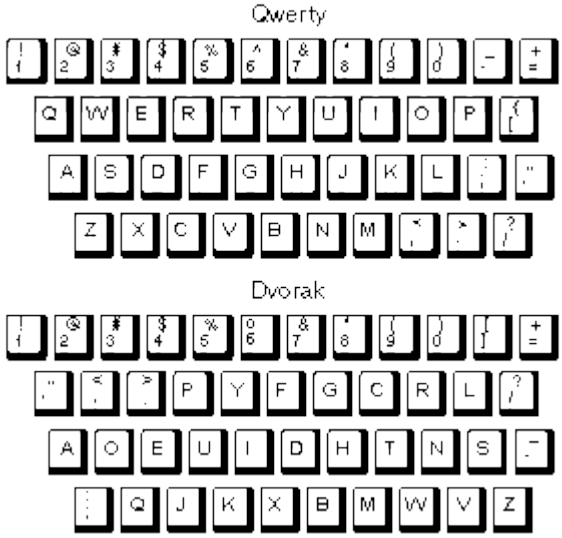
Penciptaan keyboard komputer diilhami oleh penciptaan mesin ketik yang dasar rancangannya dibuat dan dipatenkan oleh Christopher Latham pada tahun 1868. Kemudian pada tahun 1877, mesin ketik itu banyak dipasarkan oleh perusahaan Remington [1].

Keyboard komputer pertama disesuaikan dari kartu pelubang (*punch card*) dan teknologi pengiriman tulisan jarak jauh (*teletype*). Pada tahun 1946, ENIAC menggunakan pembaca kartu pembuat lubang (*puched card reader*) sebagai alat *input* dan *output.*

2.5.2. Jenis *Keyboard*

Dari sisi *layout* (tata letak) tombol, secara umum terdapat dua tipe *keyboard*, yaitu QWERTY dan DVORAK. Namun, sampai sekarang, *keyboard* qwerty mesih mendominasi dalam masalah alat *input* komputeri. Adapun penamaan qwerty berdasarkan enam huruf yang pertama kali tersusun di papan

ketik qwerty sedangkan dvorak berdasarkan nama penemunya, yaitu August Dvorak. Gambar kedua jenis *keyboard* tersebut dapat dilihat pada Gambar 2.7 [1].



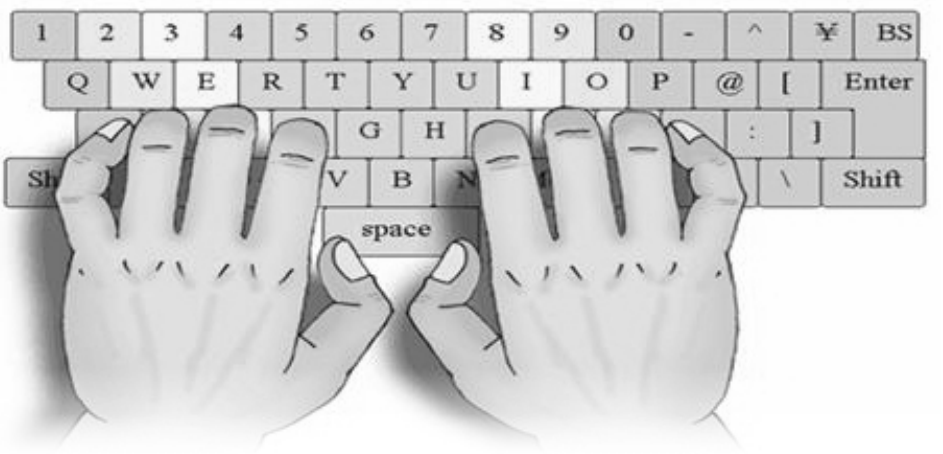
Gambar 2.7. *Keyboard* Qwerty dan Dvorak.

2.5.3. Penempatan Jari pada *Keyboard* Qwerty

Mengetik sepuluh jari merupakan kegiatan yang harus dilatih secara rutin. Tidak perlu mengeluarkan usaha yang keras dalam menghafal posisi tombol-tombol pada *keyboard* sebab kemampuan ini akan muncul dengan sendirinya karena kebiasaan jari tangan dalam menekan tombol *keyboard.*

Sebelum mulai mengetik sepuluh jari pada *keyboard* qwerty, langkah pertama yang harus dilakukan adalah menempatkan delapan jari tangan pada delapan tombol pangkal pada *keyboard* qwerty. Tombol pangkal tersebut terdiri atas A, S, D, F, J, K, L, dan ; (titik-koma). Selain itu, tombol pangkal bagian *numeric keypad* pada *keyboard* standar secara umum terdiri atas 4, 5, dan 6 sedangkan pada keyboard laptop terdiri dari 1, 2, dan 3. Dengan berpatokan

kepada tombol pangkal, tombol-tombol yang lain dapat diraih dengan menggeser jari beberapa sentimeter apabila sudah terbiasa. Setiap *keyboard* qwery biasanya memiliki tonjolan pada tombol pangkalnya. Hal ini memudahkan pengetik untuk menempatkan jarinya saat pengetikan. Tonjolan tersebut biasanya terdapat pada tombol F, J, dan 5 untuk bagian *numeric keypad.* Jadi, seorang pengetik dapat dengan mudah merasakan posisi jarinya tanpa melihat langsung ke papan ketik. Ilustrasinya dapat dilihat pada Gambar 2.8 sedangkan uraian yang lebih rincinya terdapat pada Tabel 2.1 dan Tabel 2.2 [1].



Gambar 2.8. Penempatan delapan jari pada *keyboard* qwerty.

Tabel 2.1. Penempatan jari pada *keyboard* qwerty.

Tangan	Jari	Pangkal	Tombol-tombol yang ditekan
Kiri	Kelingking	A	` 1 [TAB] q a [LSHIFT] z
	Manis	S	2 W S X
	Tengah	D	3 E D C
	Telunjuk	F	4 5 R T F G V B
	Ibu Jari	[SPASI]	[SPASI] (alternatif)
Kanan	Ibu Jari	[SPASI]	[SPASI] (utama)
	Telunjuk	J	6 7 Y U H J N M
	Tengah	K	8 I K ,
	Manis	L	9 O L .
	Kelingking	;	0 - = [BS] P [ ] \ ; ' [ENTER] / [RSHIFT]

Tabel 2.2. Penempatan jari pada keyboard laptop bagian *numeric keypad.*

Jari Tangan Kanan	Pangkal	Tombol-tombol yang ditekan
Telunjuk	1	0 1 4 7
Tengah	2	2 5 8
Manis	3	. 3 6 9
Kelingking	-	+ - * / [ENTER]

2.6. Pengertian Dasar Sistem

Suatu sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat menentukan dalam mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan.

Dalam mendefinisikan sistem terdapat dua kelompok pendekatan sistem, yaitu sistem yang lebih menekankan pada prosedur dan elemennya. Prosedur didefinisikan sebagai suatu urut-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa yang harus dikerjakan, siapa yang mengerjakan, kapan dikerjakan, dan bagaimana mengerjakannya (Gerald. J., 1991). Penganut pendekatan elemen adalah Davis (1985) yang mendefinisikan sistem sebagai bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud. Sedangkan Lucas (1989) mendefinisikan sistem sebagai suatu komponen atau variabel yang terorganisir, saling berinteraksi, saling bergantung, satu sama lain, dan terpadu. Sebuah sistem mempunyai tujuan atau sasaran. McLeod berpendapat, sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Begitu pula Robert G. Murdick (1993), mendefinisikan sistem sebagai seperangkat elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan bersama. Pendekatan sistem yang lebih menekankan pada prosedur didefinisikan bahwa sistem yaitu suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Gerald. J. 1991) [8].

Lebih lanjut pemahaman tentang sistem pertama kali dapat diperoleh dari pengertian dan definisinya. Dengan demikian definisi ini akan mempunyai peranan yang sangat penting dalam melakukan pendekatan terhadap sistem yang akan dianalisis. Pendekatan sistem yang merupakan kumpulan dari komponen

atau elemen-elemen atau subsistem-subsistem merupakan definisi yang lebih luas dibandingkan pendekatan sistem yang lebih menekankan pada prosedurnya. Definisi ini lebih banyak diterima karena pada kenyataannya suatu sistem memang terdiri dari subsistem-subsistem.

2.6.1. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat–sifat tertentu antara lain sebagai berikut: Pemetaan karakteristik sistem dapat dilihat pada Gambar 2.9 [8].

1. Komponen-komponen Sistem (*Components*)

   Komponen-komponen Sistem merupakan salah satu karakteristik sistem yang berupa sub sistem atau gagasan sistem.

2. Batas Sistem (*Boundary*)

   Batas Sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan lainnya.

3. Lingkungan Luar Sistem (*Environment*)

   Segala sesuatu yang berada di luar batas dari sistem yang mempengaruhi operasi baik itu menguntungkan operasi sistem yang merupakan energi dari sistem yang harus dipelihara atau ditahan sehingga tidak mengganggu atau merusak sistem.

4. Penghubung Sistem (*Interface*)

   Penghubung Sistem merupakan media penghubung antara satu sub sistem dengan sub sistem lainnya.

5. Masukan Sistem (*Input*)

Masukan sistem adalah data yang dimasukkan ke dalam sistem yang diproses untuk mendapat satu kesatuan atau informasi.

6. Keluaran Sistem (*Output*)

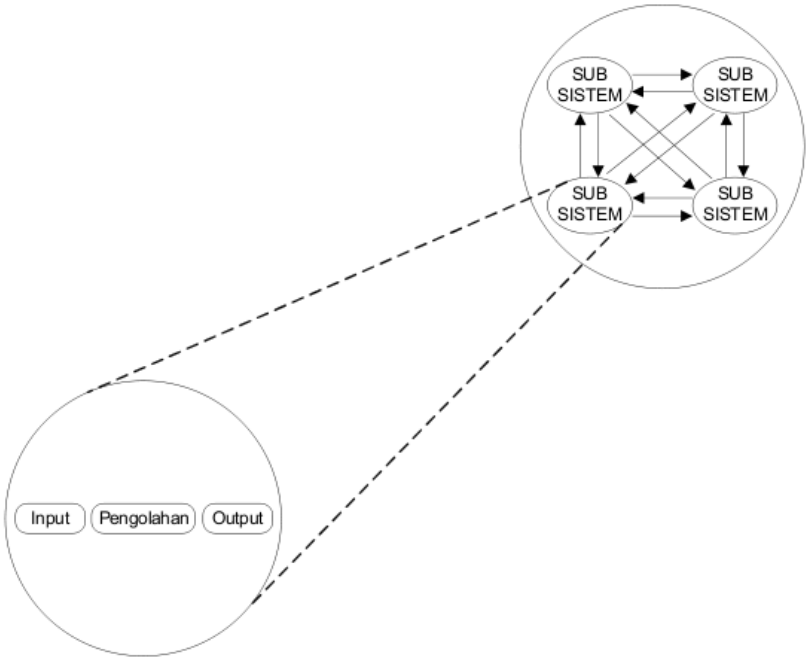
Keluaran dari sistem adalah data yang diolah serta diklasifikasikan menjadi keluaran yang berguna atau menjadi informasi yang dibutuhkan.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah suatu masukan menjadi keluaran yang dibentuk.

8. Sasaran Sistem (*Goal*)

Suatu sistem mempunyai maksud tertentu yaitu tujuan atas sasaran, dimana yang menentukan sekali masukan serta keluaran sistem yang mengena pada sasaran atau tujuan yang dimaksudkan.



Gambar 2.9. Karakteristik Sistem.

2.6.2. Klasifikasi Sistem

Dari berbagai sudut pandang, sistem dapat diklasifikasikan menjadi beberapa bagian yaitu [8]:

1. Sistem abstrak dan sistem fisik

   Sistem abstrak merupakan sistem yang tidak bisa dilihat secara mata biasa dan biasanya sistem ini berupa pemikiran atau ide-ide. Contoh dari sistem abstrak ini adalah filsafat. Sistem fisik merupakan sistem yang bisa dilihat secara mata biasa dan biasanya sering digunakan oleh manusia. Contoh dari sistem fisik ini adalah sistem akuntansi, sistem komputer dan sebagainya.

2. Sistem alamiah dan sistem buatan

Sistem alamiah merupakan sistem yang terjadi karena pengaruh alam. Misalnya sistem rotasi bumi, sistem grafitasi dan sebagainya. Sistem buatan merupakan sistem yang dirancang dan dibuat oleh manusia. Misalnya sistem pengolahan gaji.

3. Sistem tertutup dan sistem terbuka

Sistem tertutup merupakan sistem yang tidak berhubungan dengan bagian luar sistem dan biasanya tidak terpengaruh oleh kondisi diluar sistem. Sedangkan sistem terbuka merupakan sistem yang berhubungan dengan bagian luar sistem.

2.7. Pemodelan Analisis

Selama bertahun-tahun telah banyak diusulkan untuk pemodelan analisis tetapi saat ini, ada dua yang mendominasi pemodelan analisis. Yang pertama, analisis terstruktur yang merupakan metode pemodelan klasik dan yang kedua, analisis berorientasi objek.

Analisis terstruktur adala aktivitas pembangunan model. Dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional, kita dapat menciptakan model yang menggambarkan muatan dan aliran informasi (data dan kontrol), membagi sistem secara fungsional dan secara *behavioral*, dan menggambarkan esensi dari apa yang harus dibangun. Analisis terstruktur bukan merupakan metode tunggal yang diaplikasikan secara konsisten oleh semua yang

menggunakannya. Analisis terstruktur lebih merupakan campuran yang berkembang selama lebih dari 20 tahun [9].

2.7.1. Diagram Aliran Data

Diagram aliran data / *data flow diagram* (DFD) merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan menggunakan diagram aliran data adalah memudahkan pemakai atau *user* yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan [8].

2.7.1.1. Diagram Konteks

Diagram konteks (*context diagram*) adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan *level* tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Diagram konteks akan memberi gambaran tentang keseluruhan sistem. Dalam diagram konteks hanya ada satu proses serta tidak boleh ada *store* dalam diagram konteks.

2.7.1.2. Diagram Rinci

Diagram rinci (*level diagram*) adalah diagram yang menguraikan proses apa yang ada dalam diagram *zero* atau diagram *level* di atasnya.

2.7.1.3. Spesifikasi Proses (*Process Spectification*)

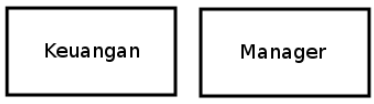
Setiap proses (*bubble*) di DFD harus memiliki spesifikasi proses (*process specification*). Tanpa ini kita tidak akan mengetahui apa yang akan terjadi di dalam proses (*bubble*) tersebut.

2.7.2. Elemen-elemen Dasar dari Diagram Aliran Data

Elemen dasar pada DFD meliputi kesatuan luar / entitas eksternal (*external entity*), arus data (*data flow*), proses (*process*). dan simpanan data (*data store*) [8].

2.7.2.1. Entitas Luar

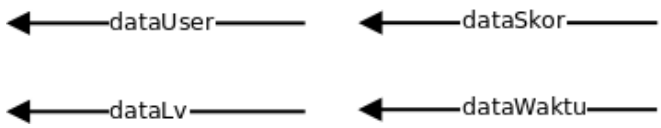
Entitas eksternal merupakan sesuatu yang berada di luar sistem tetapi memberikan data ke dalam sistem atau memberikan data dari sistem. Disimbolkan dengan suatu kotak notasi. Entitas eksternal tidak termasuk bagian dari sistem. Bila sistem informasi dirancang untuk satu bagian (departemen), maka bagian lain yang masih terkait menjadi entitas eksternal. Entitas eksternal dapat dimodelkan seperti pada Gambar 2.10.



Gambar 2.10. Contoh entitas eksternal.

2.7.2.2. Arus Data

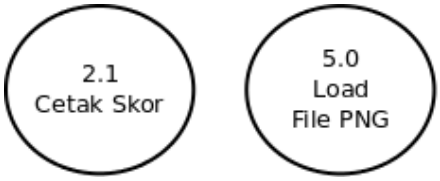
Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir di antara proses dan *data store* serta menunjukkan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem. Contoh bentuk arus data dapat dilihat pada Gambar 2.11.



Gambar 2.11. Contoh arus data.

2.7.2.3. Proses

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data ke luar. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering pula disebut dengan *bubble*. Salah satu contoh model dapat dilihat pada Gambar 2.12.



Gambar 2.12. Contoh model proses.

2.7.2.4. Simpanan Data (*Data Store*)

Simpanan data merupakan tempat penyimpanan data pengikat data yang ada dalam sistem. Data store dapat disimbolkan dengan sepasang dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke *database.* Contoh simbol ini dapat dilihat pada Gambar 2.13.



Gambar 2.13. Contoh simbol *data store.*

2.7.3. Kamus Data

Kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan, dan proses.

Kamus data sering disebut juga dengan sistem *data dictionary* yang merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap analisis, kamus data 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.

2.8. Perangkat Lunak

Perangkat lunak (*software*) adalah perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan. Di dalam program tersebut terdapat struktur data yang memungkinkan program memanipulasi informasi secara proporsional. Untuk mempermudah penggunaannya, harus disertakan dokumen yang menggambarkan operasi dan kegunaan program [9].

2.8.1. Sistem Operasi

Sistem operasi (*operating system* / OS) merupakan program yang ditulis untuk mengendalikan dan mengkoordinasi kegiatan dari sistem komputer. OS berfungsi seperti manajer di dalam perusahaan, yaitu bertanggung jawab, mengendalikan, dan mengkoordinasikan semua operasi kegiatan perusahaan secara efisien dan efektif [14].

2.8.2. *File* PNG

PNG (*Portable Network Graphics*) adalah salah satu format penyimpanan citra (gambar) yang menggunakan metode pemadatan yang tidak menghilangkan bagian dari citra tersebut (*lossless compression*). Format PNG ini diperkenalkan untuk menggantikan format penyimpanan citra GIF. Selain itu, citra dengan format PNG mempunyai faktor kompresi yang lebih baik dibandingkan dengan GIF (5% - 25% lebih baik dibanding format GIF) [16].

2.8.3. *File* Ogg

Ogg merupakan format *file* suara yang dipelihara oleh Xiph.org Fondation dan disebar-luaskan ke publik secara bebas (*free*). Format ini dirancang untuk membuat efisien multimedia digital *high quality* dalam masalah *streaming* dan manipulasi [17].

2.8.4. Bahasa C

Bahasa pemrograman adalah suatu kumpulan kata (perintah) yang siap digunakan untuk menulis suatu kode program sehingga kode-kode program yang kita tulis tersebut akan dapat dikenali oleh kompilator yang sesuai. Dalam dunia pemrograman, kata-kata tersebut sering dikenal dengan istilah *keyword* (terkadang disebut *reserved word*) [10].

Sekarang ini, banyak sekali bahasa pemrograman yang dapat digunakan untuk mengembangkan suatu perangkat lunak, diantaranya C, C++, Pascal, dan Java. Bahkan untuk pembuatan pemrograman visual pun telah banyak tersedia perangkat lunak seperti C++ Builder, Delphi, Jbuilder, Visual C++, dan yang lainnya.

Bahasa C merupakan bahasa pemrograman yang berkekuatan tinggi (*powerful*) dan fleksibel yang telah banyak digunakan oleh para *programmer* profesional untuk mengembangkan program-program yang sangat bervariasi dalam berbagai bidang.

Lahirnya bahasa pemrograman diawali oleh terbentuknya bahasa assembly yang dikembangkan oleh IBM dalam tahun 1956-1963. Bahasa ini termasuk

dalam bahasa tingkat rendah (low level language). Pada tahun 1957, sebuah tim yang dipimpin oleh John W. Backus berhasil mengembangkan sebuah bahasa pemrograman baru yang lebih diarahkan untuk proses analisa numerik. Bahasa pemrograman tersebut dinamai dengan bahasa FORTRAN (*Formula Translation*). Setahun kemudian, yaitu pada tahun 1958, para ilmuwan komputer dari Eropa dan Amerika yang tergabung dalam sebuah komite menciptakan bahasa pemrograman baru yang lebih bersifat struktural dan dinamakan dengan bahasa ALGOL (*Algorithmic Language*). Kemudian pada tahun 1964, IBM kembali menciptakan bahasa pemrograman baru dengan nama PL/I (*Programming Language 1*) yang lebih ditujukan untuk keperluan bisnis dan penelitian.

Tahun 1969, Laboratorium Bell AT&T di Murray, New Jersey, menggunakan bahasa assembly untuk mengembangkan sistem operasi Unix yang bertujuan untuk membuat program antarmuka yang bersifat *programmer friendly*. Setelah Unix berjalan, lahirlah bahasa pemrograman baru yang ditulis oleh Martin Richards dengan nama bahasa BCPL (*Basic Combined Programming Language*). Kemudian pada tahun 1970, seorang pengembang sistem dari laboratorium tersebut yang bernama Ken Thompson membuat bahasa B yang akan digunakan untuk menulis ulang sistem operasi Unix. Nama B ini konon diambil dari huruf pertama dalam kata BCPL. Karena alasan bahwa bahasa B masih berkesan lambat, maka pada tahun 1971, seorang pengembang sistem bernama Dennis Ritchie yang juga bekerja di laboratorium yang sama, menciptakan bahasa baru dengan nama C yang bertujuan untuk menulis ulang dan menutupi kelemahan-

kelemahan yang ada pada sistem operasi Unix sebelumnya. Menurut sumber yang ada, nama C ini juga konon diambil dari huruf kedua dalam kata BCPL.

Sejak itu bahasa C terus digunakan untuk memelihara sistem operasi Unix. Sampai akhirnya, pada tahun 90-an, bahasa C ini digunakan untuk mengembangkan sistem operasi Windows dan sekarang ini digunakan untuk mengembangkan sistem operasi Linux. Selain untuk menulis program yang merupakan *embedded system,* di kalangan industri hiburan, bahasa C juga banyak digunakan dalam mengembangkan perangkat lunak untuk permainan (*game*). Hal-hal inilah yang menyebabkan bahasa C menjadi bahasa yang sangat populer di kalangan industri perangkat lunak.

2.8.5. Pustaka SDL

SDL (Simple DirectMedia Layer) adalah pustaka (*library*) pemrograman untuk membuat aplikasi multimedia di berbagai sistem operasi. Dengan menggunakan SDL, *programmer* dapat mengakses layar, suara, papan ketik, *joystick*, *hardware* 3D dan 2D *framebuffer* dengan menggunakan cara yang sama di berbagai sistem operasi. Kelebihan lain dari SDL adalah dapat digunakan dengan berbagai bahasa pemrograman [11].

SDL pertama kali ditulis oleh Sam Lantinga pada tahun 1998. Sam menulis SDL karena dia hendak membuat permainan komputer yang dapat berjalan di Windows dan Macintosh. Setelah beberapa game dibuat, Sam menulis SDL untuk sistem operasi lain seperti BeOS dan Linux. Hal ini disebabkan Sam ingin menjalankan permainan komputer Doom di berbagai macam sistem operasi.

2.8.6. Bloodshed Dev-C++

IDE (*Integrated Development Environment*) merupakan program komputer yang memiliki fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Sebuah IDE setidaknya memiliki fasilitas:

1. Editor, yaitu fasilitas untuk menuliskan kode sumber dari *software.*
2. *Compiler*, yaitu untuk mengkompilasi kode sumber ke bentuk biner.
3. *Linker*, yaitu fasilitas penyatu beberapa *library* atau *file* objek.
4. *Debugger*, yaitu fasilitas untuk mendeteksi kesalahan program.

Bloodshed Dev-C++ merupakan IDE yang khusus ditujukan untuk *programmer* C/C++. Program ini menggunakan sebagian fasilitas dari MinGW (IDE) sebagai *compiler*-nya. Dev-C++ dapat dikombinasikan juga dengan Cygwin (IDE) atau yang berbasis GCC (*compiler*) lainnya [12].

2.8.7. GIMP

GIMP adalah program aplikasi yang berguna untuk membuat dan memanipulasi foto digital yang berjalan di berbagai *platform*. GIMP merupakan kependekan dari *GNU Image Manipulation Program.* GIMP ini memiliki kesamaan dengan Photoshop tetapi GIMP lebih sering ditemukan di *platform* Linux sedangkan Photoshop di *platform* Windows [18].

2.8.8. Diagram Editor Dia

Editor diagram yang cukup dikenal di sistem operasi linux adalah Dia. Dia dibangun dari pustaka gtk+ yang berlisensi GPL. Dia merupakan program yang terinspirasi dari program di Windows, yaitu Visio. *Output* yang dihasilkan oleh aplikasi Dia bisa berupa diagram alir data, diagram relasi, diagram UML, flowchart, diagram jaringan, dan sebagainya yang berguna dalam pemodelan analisis [19].