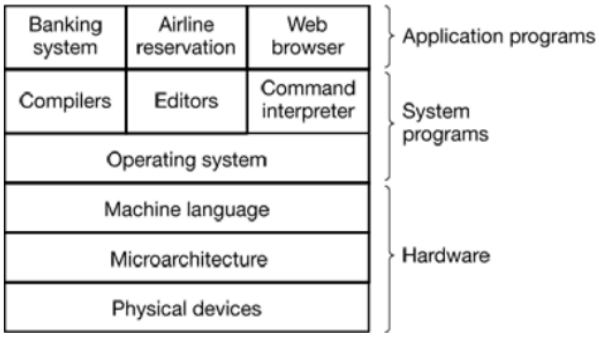
BAB II
TINJAUAN PUSTAKA

II.1 Sistem Operasi

Sistem operasi adalah perangkat lunak yang mengontrol, mengelola perangkat keras dan menyediakan program pengguna dengan antarmuka. Secara teknis Sistem Operasi disebut Kernel. Kernel adalah komponen sentral dari sistem operasi, mengatur hal-hal seperti *interrupt handler* (untuk menyediakan layanan interupsi), process scheduler(membagi-bagi proses dalam prosesor), memory management, I/O, dan sebagainya. Atau dengan kata lain, ia adalah jembatan antara hardware dengan software. Penempatan sistem operasi ditunjukan pada Gambar II.1 [**3**]



Gambar II.1 Penempatan Sistem Operasi [3**]**

Sistem operasi menghubungkan, mengelola antara sistem program, dan perangkat keras. Sistem operasi memiliki fungsional tertentu diantaranya dalam segi kenyamanan, sistem operasi menjadikan penggunaan terhadap komputer menjadi lebih nyaman digunakan, dari segi efeisen memungkinkan sumber daya sistem komputer untuk digunakan secara efisien dan memiliki izin pengembangan yang efektif, pengujian, dan pengenalan fungsi sistem baru tanpa mengganggu layanan sistem yang ada. Beberapa layanan sistem operasi adalah *program development*, eksekusi program pengaksesan ke perangkat I/O, Akses kontrol berkas dan akses sistem.

II.2 Macam-Macam Desain Kernel

Kernel merupakan bagian utama untuk mengendalikan sistem, Macam-macam desain kernel [**3**]:

1. Monolithic kernel mengintegrasikan banyak fungsi di dalam kernel dan menyediakan lapisan abstraksi perangkat keras secara penuh terhadap perangkat keras yang berada di bawah sistem operasi. Contoh dari kernel berjenis monolithic yang paling terkenal adalah Unix, Linux. Kernel monolithic mempunyai *size* yang lebih besar.
2. Micro kernel Mikrokernel menyediakan sedikit saja dari abstraksi perangkat keras dan menggunakan aplikasi yang berjalan di atas nya yang disebut dengan *serve* untuk melakukan beberapa fungsionalitas lainnya. Adapun servis dasar yang dimiliki yaitu manajemen pengalamatan memori, manajemen proses/*thread*, dan *inter-process communication*. Contoh dari desain kernel mikro ini adalah Minix.

II.3 Struktur Sistem Operasi

Sistem operasi modern mempunyai stuktur komponen sebagai berikut [**4**]:

1. Manajemen Proses
2. Manajemen Memori Utama
3. Manajemen *Secondary Storage*
4. Manajemen Sistem I/O
5. Manajemen Sistem Berkas
6. Sistem Proteksi
7. Jaringan
8. *Command Interpreter Sistem*

1. Manajemen Proses

Proses adalah keadaan ketika sebuah program sedang di eksekusi. Sebuah proses membutuhkan beberapa sumber daya untuk menyelesaikan tugasnya. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan managemen proses seperti [**4**] :

1. Pembuatan dan penghapusan proses pengguna dan sistem proses.
2. Menunda atau melanjutkan proses.
3. Menyediakan mekanisme untuk proses sinkronisasi.

2. Manajemen Memori Utama

Memori Utama berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. Memori utama termasuk tempat penyimpanan data yang sementara *(volatile),* artinya data dapat hilang begitu sistem dimatikan.

Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen memori seperti [**4**]:

1. Menjaga track dari memori yang sedang digunakan dan siapa yang menggunakannya.
2. Memilih program yang akan di-*load* ke memori.
3. Mengalokasikan dan meng-dealokasikan ruang memori sesuai kebutuhan.

3. Manajemen Secondary Storage

Penyimpanan sekunder (*secondary storage*) adalah sarana penyimpanan yang berada satu tingkat di bawah memori utama sebuah komputer dalam hirarki memori. Tidak seperti memori utama komputer, penyimpanan sekunder tidak memiliki hubungan langsung dengan prosesor melalui bus, sehingga harus melewati I/O.

Sarana penyimpanan sekunder memiliki ciri-ciri umum sebagai berikut [**4**] : *Non volatile* (tahan lama), tidak berhubungan langsung dengan bus CPU. lambat dan harganya murah. Contoh nya seperti hardisk.

4. Manajemen Sistem I/O

Salah satu fungsi utama sistem operasi adalah mengatur operasi Masukan/Keluaran (M/K) beserta perangkatnya. Sistem operasi harus dapat memberikan perintah ke perangkat-perangkat tersebut, menangkap interupsi, dan

menangani error/kesalahan yang terjadi. Selain itu, sistem operasi juga menyediakan fasilitas antarmuka *(interface)* antara perangkat-perangkat tersebut dengan keseluruhan sistem yang ada.

Komponen Sistem Operasi untuk sistem I/O [**4**]:

1. *Buffer*: menampung sementara data dari/ ke perangkat I/O.
2. *Spooling*: melakukan penjadualan pemakaian I/O sistem supaya lebih efisien (antrian dsb.).
3. Menyediakan *driver* untuk dapat melakukan operasi perangkat keras I/O tertentu.

5. Manajemen Sistem Berkas

suatu sistem untuk mengetahui bagaimana cara menyimpan data dari *file* tertentu dan organisasi *file* yang digunakan. Berkas dapat mempunyai struktur yang bersifat hirarkis (direktori, volume, dll.). Sistem operasi bertanggung-jawab [**4**]:

1. Pembuatan dan penghapusan berkas.
2. Pembuatan dan penghapusan direktori.
3. Mendukung manipulasi berkas dan direktori.
4. Memetakan berkas ke *secondary storage.*
5. *Backup* berkas ke media penyimpanan yang permanen (*non-volatile*).

6. Sistem Proteksi

Sistem proteksi merupakan mekanisme proteksi yang harus dapat membedakan antara penggunaan yang sudah diberi izin dan yang belum dan menentukan kontrol yang akan dikenakan [**4**].

7. Jaringan

Sistem terdistribusi adalah sekumpulan prosesor yang tidak berbagi memori atau clock. Tiap proses mempunyai memori sendiri. Proses-proses tersebut terhubung melalui jaringan komunikasi Sistem terdistribusi menyediakan akses pengguna ke bermacam sumber-daya sistem. Akses tersebut menyebabkan [**4**]:

1. *Computation speed-up*.
2. *Increased data availability*.
3. *Enhanced reliability*.

8. Command Interpreter System

Sistem Operasi menunggu instruksi dari pengguna (*command driven*). Program yang membaca instruksi dan mengartikan *control statements* umumnya disebut *control-card interpreter*, *command-line interpreter*, dan UNIX shell. *Command-Interpreter Sistem* sangat bervariasi dari satu sistem operasi ke sistem operasi yang lain dan disesuaikan dengan tujuan dan teknologi *I/O devices* yang ada. Contohnya: CLI, Windows, *Pen-based(touch)*, dan lain-lain [**4**].

II.4 POSIX

POSIX *(Portable Operating System Interface)* merupakan standarisasi yang ditentukan oleh IEEE untuk menjaga kompatibilitas antara sistem operasi. POSIX mempunyai tujuan untuk pemeliharaan dan pengembangan antar sistem operasi. POSIX dikembangkan 1997. Adapun standarisasi yang dihasilkan adalah 19 arsip dokumen [**5**].

Setiap sistem operasi memiliki POSIX tersendiri baik Windows, DOS, Unix dan Linux. POSIX yang berlaku untuk Linux adalah POSIX.1 2008. Adapun pembahasan adalah tentang definisi dan standar Sistem Interface, perintah, Utilitas *(Real-time Services, Threads Interface*, Ekstensi *Real-time*, Keamanan *Interface*, *File System, Network File Access* dan Komunikasi jaringan Processes to Process, *Users Portabilitas Extensions*, Correction dan *Extensions*, Perlindungan, Pengendalian Utilitas dan Batch Sistem Utilitas) dan lain sebagainya [**5**].

II.5 Tabel Partisi

sebuah struktur data yang menyimpan lokasi (permulaan dan akhir) dan ukuran dari setiap partisi utama yang terkandung di dalam sebuah *harddisk*. Dalam sebuah *harddisk* yang menggunakan skema partisi MBR, tabel partisi diletakkan pada *Master Boot Record* yang disimpan di dalam sektor pertama di

dalam *harddisk*, sementara dalam sebuah *harddisk* yang menggunakan skema partisi GPT, tabel partisi diletakkan dalam sebuah larik yang berisi entri-entri partisi, yang disebut dengan GUID Partition Entry Array. MBR mendukung kapasistas *hardisk* sampai 2 TB, menggunakan pengalamatan 32 bit untuk alokasi sektor dan hanya dapat membuat 3 partisi primary dan 1 partisi extended sedangkan GPT 128 partisi dan mendukung kapasitas hardisk sampai 18 EB (Exabyte) atau 1 juta TB (Terra Byte) [**6**].

Lokasi yang ditunjukkan oleh tabel partisi adalah lokasi fisik dari sebuah partisi tertentu yang direpresentasikan dalam notasi angka silinder, head, dan sektor. Selain itu, tabel partisi juga memuat jenis sistem berkas yang digunakan oleh setiap partisi, dan informasi apakah partisi tersebut dapat melakukan booting atau tidak.

II.6 File System

File System (Sistem berkas) adalah suatu metode pengolahan berkas dalam *secondary storage.* Berkas adalah sebuah koleksi informasi berkaitan yang diberi nama dan disimpan di dalam *secondary storage*. Biasanya sebuah berkas merepresentasikan data atau program. Beberapa jenis berkas diantaranya [**4**]:

I. *Text file*. yaitu urutan dari karakter-karakter yang diatur menjadi barisan dan mungkin halaman.
II. *Source file*. yaitu urutan dari berbagai *subroutine* dan fungsi yang masing-masing kemudian diatur sebagai deklarasi-deklarasi diikuti oleh pernyataan-pernyataan yang dapat dieksekusi.
III. *Object file*. yaitu urutan dari *byte-byte* yang diatur menjadi blok-blok yang dapat dipahami oleh penghubung system.
IV. *Executable file*. adalah kumpulan dari bagian-bagian kode yang dapat dibawa ke memori dan dijalankan oleh *loader.*

Setiap berkas memiliki *attiribute* agar mempermudah untuk membedakannya. Berkas memiliki berbagai jenis *attribute* yang berbeda-beda antar sistem operasi, secara umum berkas memiliki *attribute* sebagai berikut [**4**]:

1. Nama. Nama dari berkas yang dituliskan secara simbolik adalah satu-satunya informasi yang disimpan dalam bentuk yang dapat dibaca oleh manusia pada umumnya.
2. *Identifier* . *Tag* unik yang biasanya berupa angka, yang mengidentifikasikan berkas dalam sistem berkas. Identifier ini tidak dapat dibaca oleh manusia.
3. Jenis. Informasi yang dibutuhkan sistem yang biasanya medukung bermacam-macam tipe berkas yang berbeda.
4. Lokasi. Informasi yang berisi *pointer* ke *device* dan lokasi dari berkas dalam *device* tersebut.
5. Ukuran. Ukuran dari berkas saat ini, dan mungkin ukuran maksimum berkas juga dimasukkan dalam atribut ini.
6. Waktu, tanggal dan identifikasi pengguna . Informasi yang disimpan untuk pembuatan, modifikasi terakhir dan kapan berkas terakhir digunakan. Data-data ini dapat berguna dalam proteksi, keamanan dan monitoring penggunaan berkas.

Operasi dasar berkas diantaranya adalah :

1. Membuat berkas (*Create File*).
2. Menulis sebuah berkas (*Write File*).
3. Membaca sebuah berkas (*Read File*).
4. Memposisikan sebuah berkas (*Reposition*).
5. Menghapus berkas (*Delete*).
6. Menghapus sebagian isi berkas (*Truncate*).

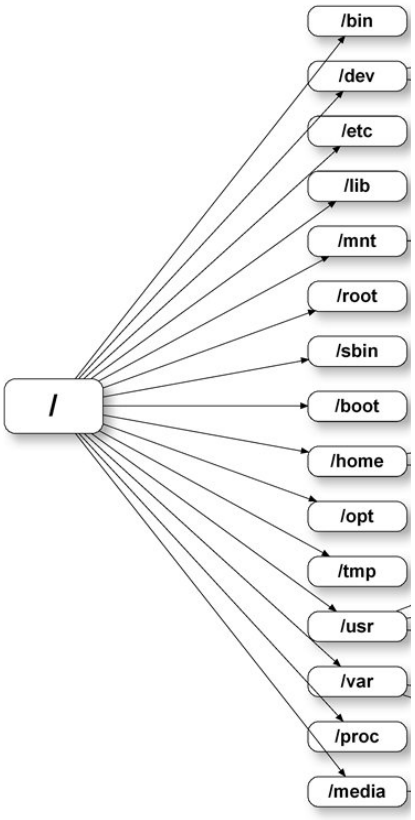
   Berkas dapat distrukturkan dalam beberapa cara, cara pertama dengan sebuah urutan bytes yang tidak terstruktur. Cara kedua dengan menggunakan *record squance* dan ketiga dengan struktur tree.

II.7 FHS

FHS (*Filesystem Hierachy Standart*) adalah seperangkat petunjuk untuk penempatan file dan direktori dibawah sistem operasi yang mirip UNIX. Tujuannya agar dapat mendukung interopabilitas aplikasi, program administrasi sistem, program pengembangan, skrip dan dapat menyatukan dokumentasi dari

sistem ini. Dengan adanya standar FHS ini, pengguna dan pengembang memiliki pedoman direktori standar apa yang dibutuhkan untuk meracik sebuah distribusi Linux yang operasional. Juga file dan pustaka, masing-masing letaknya dimana, dipandu oleh standar ini. Pengembangan standar ini pertama kali dibuat pada bulan Agustus 1993 sebagai usaha untuk menyatukan struktur file dan direktori di Linux.

Pertama dibuat dengan nama FSSTND (singkatan sama, Filesystem Hierarchy Standard), pertama kali dirilis untuk Linux 14 Februari 1994. Dalam perkembangannya, standar ini juga dibutuhkan dikomunitas BSD. Karena itu diperluas standar pada sistem operasi yang mirip UNIX (UNIX- Like) dan mengubah singkatannya menjadi FHS. Saat ini dokumen masih dirawat oleh Daniel Quinlan sendiri, dibantu Paul 'Rusty' Russel dan Daniel Yeoh. Standar terakhir adalah versi 2.3 yang diterbitkan pada tanggal 28 Januari 2004 [**7**].



Gambar II.2 FHS (Filesysten Hierachy Standart)

Berikut beberapa definisi direktori menurut standar FHS [**7**] :

1. / (Root folder) : menduduki posisi puncak di dalam hirarki, direktori ini dilambangkan dengan tanda slash (/) atau biasa disebut garis membawahi semua.
2. /bin : Direktori bin berisi perintah dasar yang dibutuhkan oleh system maupun pengguna.
3. /boot : berisi program dan data yang dibutuhkan pada saat melakukan proses booting (menjalankan) system.
4. /dev : direktori tempat file device
5. /etc : berisi file konfigurasi system
6. /home : direktori tempat menyimpan data pengguna.
7. /lib : berisi file-file library dari aplikasi yang ada di sistem.
8. /media : direktori ini berisi penyimpanan *mounting device*.
9. /mnt : direktori tempat pengaitan sistem sementara
10. /opt : berisi paket aplikasi tambahan dari pengguna.
11. /proc : direktori manajemen proses.
12. /root : direktori untuk pengguna root.

II.8 Metode LFS (Linux From Scratch)

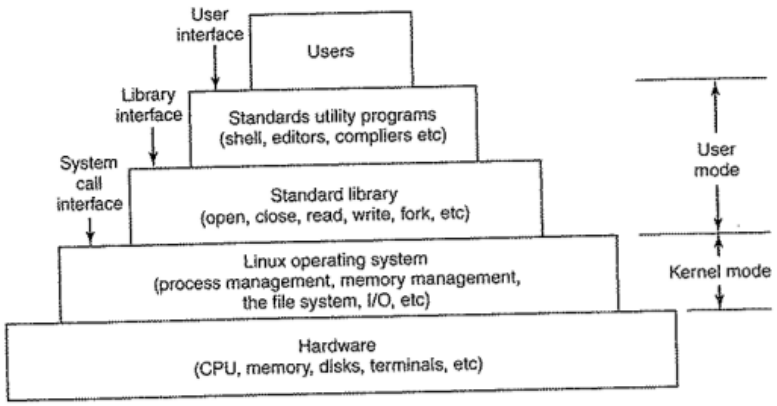
Metode LFS adalah metode pembangunan atau pembuatan distro GNU/Linux dengan cara kompilasi dan pergantian root dari dasar atau secara manual. Metode ini merupakan metode yang menggunakan sistem operasi GNU/Linux dengan penggantian *root* dengan arsitektur amd/intel x86 (32-bit) dan x86_64 (64-bit). Fungsi utama dari metode ini adalah sistem yang lebih fleksibel, aman dan pemahaman yang lebih besar tentang cara kerja internal sistem operasi berbasis Linux [**7**].

II.9 Linux

Linux merupakan sistem operasi yang dibuat oleh linus torvalds. Linux pertama kali dirilis pada tahun 1991 dengan versi 0.01. Kernel linux berjenis monolithic kernel dimana desain monolithic ini mempunyai hampir semua servis

dasar sistem operasi dari modulasi, device driver, virtual memory, scheduler, IPC, system call dan lain sebagainya [**3**].

Sistem linux dapat diilustrasikan seperti piramid, dimana masing-masing *layer* memiliki fungsional nya masing-masing, Ilustrasi ditunjukan pada Gambar II.3 Layer pada Sistem Linux



Gambar II.3 Layer pada Sistem Linux [3**].**

Program memilliki sistem panggil, menempatkan dan mengeluarkan instruksi dari mode pengguna pada mode kernel. Pada kernel, memanggil dan membagi sistem panggil pada library. Library berinteraksi pada layer atas program untuk dimunculkan ke tahap pengguna (*users*).

II.10 LSB (Linux Standar Base)

LSB (*Linux Standar Base*) adalah paket-paket atau *library* standar untuk membantu proses pembangunan sistem operasi berbasis linux. LSB yang dibutuhkan dibagi menjadi enam bagian [**7**] :

1. LSB core adalah paket utama untuk membantu pembangunan inti sistem operasi. Pada LSB core ini biasanya tersedia utilitas untuk pengolahan direktori, berkas, kompresi, dan lain sebagainya.
2. LSB C++ adalah library C++ didalamnya tersedia compiler seperti GCC.
3. LSB Desktop adalah paket yang dibutuhkan untuk membangun antar muka berbasis GUI *(Graphical User Interface)* didalamnya terdapat

library yang berhubungan dengan grafis seperti Xorg, MesaLib, Libpng, LibJpeg dan lain-lain.

4. LSB Multimedia adalah paket yang dibutuhkan untuk kebutuhan multimedia baik untuk audio dan video.
5. LSB Printing adalah paket untuk kebutuhan komunikasi perangkat keras cetak (*printer*) terhadap modul pada kernel.
6. LSB Runtime Language adalah paket untuk kebutuhan *encoding* dan bahasa.

II.11 Interaksi Manusia dan Komputer

Interaksi Manusia dan Komputer (IMK) adalah suatu disiplin ilmu yang mengkaji komunikasi atau interaksi di antara pengguna dengan sistem. Sistem yang dimaksudkan di sini tidak hanya sistem yang ada di komputer saja, tetapi juga sistem yang banyak digunakan dalam kehidupan sehari-hari, seperti kendaraan, peralatan rumah tangga, dan sebagainya. Peran utama dari HCI adalah untuk menghasilkan sebuah sistem yang mudah digunakan, aman, efektif, dan efisien [**8**].

Kunci utama IMK adalah daya guna (*usability*), yang berarti suatu sistem harus mudah digunakan, memberikan keamanan kepada pengguna, mudah dipelajari, dan sebagainya [**8**].

II.12 Analisis Tugas *(Task Analysis)*

Analisis tugas adalah bagian dari disiplin ilmu Interaksi Manusia dan Komputer. Analisis tugas merupakan inti dari pekerjaan interaksi manusia dan komputer, karena berkaitan dengan performa pekerjaan. Secara umum analisis tugas adalah proses menganalisis dan menggambarkan bagaimana manusia melaksanakan tugas atau pekerjaannya, apa saja yang dilakukan, peralatan apa saja yang digunakan dan hal-hal apa saja yang perlu diketahui [**8**].

Istilah-istilah dalam analisis tugas :

1. Sasaran (*eksternal task*) adalah suatu kondisi yang ingin dicapai manusia. Contoh : pergi ke Pasar.
2. Tugas (*internal task*) adalah suatu prosedur yang dibutuhkan untuk mencapai sasaran dengan menggunakan perangkat tertentu. Contoh : menulis perintah dengan keyboard.
3. Aksi (*action*) adalah tugas yang tidak memiliki pemecahan (*breakdown*). Contoh : menekan tombol.
4. Rencana (*method*) adalah sejumlah aksi yang disusun dalam suatu urutan.

Secara umum analisis tugas memiliki beberapa fungsional [**8**]:

1. Manual dan pengajaran
   a. Mengajarkan cara melakukan *taks.*
   b. Menyusun manual atau materi ajar.
   c. Membantu *user* menjelaskan sistem ke orang lain.
2. Menangkap kebutuhan dan merancang sistem
   a. Membantu perancangan sistem baru.
   b. Membantu perancang dalam memilih model internal untuk sistem yang diharapkan pengguna.
   c. Meramalkan penggunaan sistem baru.
3. Merancang antarmuka *detail*
   a. Mengklasifikasi tugas atau objek yang digunakan dalam perancangan menu.
   b. Menghubungkan objek dengan aksi.

Adapun contoh analisis tugas dalam membersihkan rumah adalah sebagai berikut :

1. Ambil penghisap debu.
2. Tancapkan penghisap debu ke alat penghubung listrik.
3. Bersihkan ruangan.
4. Jika kotak debu telah penuh, kosongkan.
5. Pasang kembali penghisap debu dan segala peralatan pembantunya.

Pengguna yang melakukan perlengkapan ini harus mengetahui tentang :

1. Penghisap debu.
2. Alat-alat pembantu.
3. Kotak debu.
4. Lemari.
5. Ruangan, dan sebagainya.

II.13 Teknik Analisis Tugas

Teknis analisis tugas dibagi menjadi tiga bagian, diantaranya adalah [**8**]:

1. Dekomposisi tugas,
2. Analisis berbasis pengetahuan,
3. Teknik berbasis relasi

Perbedaan antara teknik analisis tugas dan teknik yang lainnya adalah bahwa teknik analisis tugas memiliki ruang lingkup yang luas. Selain meliputi tugas-tugas yang melibatkan penggunaan komputer, analisis tugas juga memodelkan aspek-aspek dunia nyata baik yang menjadi bagian maupun tidak menjadi bagian sistem komputer. Contoh, jika dilakukan analisis tugas terhadap pekerjaan pengolahan kata maka aktifitas mengambil kertas, mengganti tinta komputer, memasukan flashdisk kiha akan menjadi bagian-bagian dari hal-hal yang tercakup di dalamnya.

II.13.1 Dekomposisi Tugas

Dekomposisi tugas memisahkan tugas ke dalam urutan sub-tugas, bertujuan untuk menjelaskan aksi yang dilakukan manusia, menstrukturkan tugas di dalam heirarki sub-tugas dan menjelaskan urutan dari sub-tugas.

HTA (*hierarchical task analysis*) merupakan metode yang ekonomis dalam pengumpulan dan pengorganisasian informasi karena analis hanya perlu mengembangkan bagian dari herarki yang dibutuhkan. Struktur heirarki HTA memungkinkan analis memfokuskan pada aspek penting task dalam konteks keseluruhan task.

Kerugian dari HTA adalah bahwa analis perlu mengembangkan pengukuran keterampilan untuk menganalisis tugas secara efektif. Teknik ini bukan prosedur

yang sederhana yang dapat diterapkan secara cepat. Keterampilan tersebut dapat diperoleh dengan cepat mealalui latihan [**8**].

Fokus analisis hierarki tugas adalah penggunaan teks dan diagram untuk menunjukan hierarki dan perencanaan untuk menjelaskan urutan. dalam rangka membersihkan rumah[4]. :

1. Keluarkan penghisap debu.
2. Sesuaikan semua alat yang harus ditancapkan.
3. Bersihkan ruangan.
   3.1 Bersihkan ruangan utama.
   3.2 Bersihkan ruangan tamu.
   3.3 Bersihkan kamar tidur
4. Jika kotak debu nya sudah penuh, kosongkan.
5. Letakan penghisap debunya dan segala peralatan pembantunya.

Perencanaan :

Rencana 0: kerjakan 1-2-3-4-5 dalam urutan

Ketika kotak debu penuh, kerjakan 4.

Rencana 3 : kerjakan sembarang dari 3.1, 3.2, atau 3.3 dalam sembarang order tergantung pada ruang mana yang butuh dibersihkan.

HTA merupakan suatu tata bahasa untuk menjelaskan urutan-urutan tugas yang diakukan dengan menguraikan kalimat berbentuk huruf-huruf, kata benda, kelompok kata benda dan lain sebagainya.

HTA juga dapat dibuat dalam bentuk diagram seperti berikut :



Gambar II.4 Diagram HTA

II.13.2 Analisis Berbasis Pengetahuan

Analisis berbasis pengetahuan dimulai dengan mendaftar semua objek dan aksi yang terlibat dalam tugas dan kemudian membangun taksonominya. Hal ini mirip dengan deskripsi hierarki yang dilakukan pada bidang biologi. Hewan digolongkan menjadi invertebrata dan vertebrata. Hewan vertebrata adalah ikan, burung, reptil, amfibi mamalia dan sebagainya. Tujuannya adalah untuk mengetahui dan memahami pengetahuan yang dibutuhkan untuk melaksanakan tugas dan dapat digunakan untuk membantu membuat materi pengajaran dan menilai jumlah pengetahuan pada tugas yang berbeda [**8**].

Ada tiga tipe notasi dari taksonomi, yaitu :

1. XOR – taksonomi normal : objek ada dalam satu-satunya cabang atau sebuah objek hanya merupakan bagian dari satu kategori.
2. AND – objek harus ada pada keduanya. Digunakan ketika suatu objek terdiri dari beberapa ketegori untuk mempresentasikan klasifikasi jamak.
3. OR – kasus terlemah: Dapat saja pada satu, banyak atau tak ada cabang, digunakan jika objek ,erupakan bagian dari suatu atau lebih kategori.

TAKD (*Task Analysis for Knowledge Base*) mempunyai aturan keunikan yang menuntut TDH (*Task Descriptive Hierarchy*) yang lengkap dapat membedakan dua objek yang spesifik. Bentuknya dapat dilihat dalam Tabel II.1 berikut [4].:

Tabel II.1 Tabel TDH (*Task Descriptive Hierarchy*)

1.	*Kitchen item AND*
2.	*/____shape XOR*
3	*/____|____dished mixing bowl, casserole, saucepan,*
4.	*/____| soup bowl, glass.*
5.	*/____|____flat plate, chopping board, frying pan*
6.	*/____function OR*
7.	*{____preparation mixing bowl, plate,*
8.	*chopping board.*
9.	*{____cooking frying pan, casserole,*
10.	*saucepan*
11.	*{____dining XOR*
12.	*|____for food plate, soup bowl,*
13.	*casserole*
14.	*|____for drink glass*

Keterangan :

1. "/", digunakan untuk merepresentasikan percabangan.
2. "|", digunakan XOR
3. "{",digunakan untuk OR.

Setelah itu setiap objek dapat direpresentasikan dengan jejak khusus dalam hierarki yang disebut *Knwoladge Representation Grammer* (KRG). KRG dibuat menggunakan "/" cabang AND. "()" untuk XOR dan "{}" untuk OR sebagai contoh mereferensikan flate sebagai :

Kitchen item/shape(flat)/function{preparation,dining(for food)}/

KRG diatas dibaca menjadi :

Kitchen item whose shaped is flat AND its function is preparation OR for food

Peralatan dapur yang bentuknya datar dan fungsinya untuk persiapan ATAU untuk makanan.

Aturan keunikan tidak harus selalu dipenuhi secara kaku/ terutama pada hierarki sederhana yang tidak terlalu kompleks.yang mengandung percabangan.

II.13.3 Teknik Berbasis Relasi Entitas

Teknik berbasis relasi entitas biasanya berasosiasi dengan basis data pada model database entitas, mewakili sistem contoh tabel dan atribut pada analisis tugas, ,enekankan pada objek, aksi dan hubungan diantaranya, mirip dengan analisis berbasis objek tetapi mengikut-sertakan entitas non-komputer dan penekanan pada pemahaman domain, bukan implementasi [**8**].

1. Objek

Objek dibagi menjadi tiga bagian :

1. Objek Kongkret : Sesuatu yang sederhana: Cangkir kopi, blender.
2. Aktor : memasak, makan malam.
3. Objek KompositL staff dapur, resep.

2. Aksi

Aksi merupakan mengubah kondisi sesuatu (*patient, diubah oleh aksi*) menggunakan sesuatu *(instrument, alat yangdigunakan aksi)* dan dilakukan oleh seseorang *(agent).* Contoh :

Chef *(agent)* mencampur *(action)* resep *(patient)* dengan sendok *(instument).*

Agen bertanggung jawab untuk setiap aksi , karena cara yang baik untuk menandakan bahwa aktor mengali sistem. Suatu pesan adalah kasus spesial pada suatu aksi. Sebagai contoh: Koki utama memberi tahu Michael cara memotong wortel. Ada dua kemungkinan aksi :

1. Koki pertama memberi pesan ke michael.
2. Michael memotong wortel.

Contoh E/R objek dan aksi pada

Tabel II.2 sebagai berikut :

Tabel II.2 E/R Objek

1.	Object chef human actor
2.	Actions
3.	C1 : Memberi tahu Michael membuat fairy
4.	cake
5.	Object Michael human actor
6.	Actions
7.	M1 : Mencampur semua resep.
8.	M2 : Meletakan campuran di oven.
9.	M3 : Meletakan cake di rak pendingin
10.	Object Oven simple
11.	Attributes
12.	Temperature 0-220 degress
13.	Object fairy cake simple
14.	Actions
15.	F1 : Mengembang
16.	Events
17.	Ev1: Bakar
18.	Relations : Object-object
19.	Lokasi(Oven,Dapur)
20.	Relations : Action-object
21.	Patient (C1, Michael)
22.	Patient (M1, Michael)
23.	Relations : Object-event
24.	Penyebab (M2,EV1)

3. Kejadian

Kejadian (Event) adalah saat sesuatu berlangsung atau terjadi.

1. Kinerja aksi : “Galuh memanen kacang kedelai”.

2. Kejadian spontan : ‘biji labu berkecambah’ dan ‘kelembaban turun dibawah 25%’ (agen tidak bertanggung jawab atas peristiwa yang terjadi).
3. Kejadian berdasarkan waktu: ‘pada tengah malam’.

4. Hubungan

Hubungan (*relations*) merupakan suatu hubungan antara objek, aksi dan kejadian. Relasi antara komponen-komponen tersebut adalah sebagai berikut :

1. Object-object
   Social : Sam adalah bawahan vera.
   Spatial : Pompa 3 adalah untuk rumah kaca.
2. Action-object
   Agen yang berhubungan dengan objek.
3. Action-events
   Bersifat sementara dan menyebabkan, contoh ‘Sam memanen wortel karena vera memberitahukannya’.
4. Temporal relations
   Menggunakan HTA atau notasi dialog.
   Menunjukan urutan tugas,

II.14 Penggunaan Hasil Analisis Tugas

Keluaran analisis tugas merupakan perincian dari tugas yang dilakukan orang, teknik yang mereka gunakan, alat yang mereka gunakan serta rencana dan urutan aksi untuk melaksanakn tugas tersebut.

1. Manual dan pengajaran : Struktur hierarki yang dimiliki HTA dapat digunakan untuk menyusun manual atau bahan pengajaran.
2. Pendefinisian Kebutuhan dan Perancangan Sistem akan membantu pada dua hal yaitu :
   a. Objek dan tugas apa saja yang ada di sistem lama yang akan diakomodasi di sistem baru.

b. Fitur apa yang akan diperbaharui, apakah akan mengotomasi seluruh tugas atau fungsi atau tugas spesifik tertentu.

3. Perancangan detail interface. Taksonomi tugas atau objek dapat digunakan untuk merancang menu. Tugas-tugas utama dapat dijadikan menu utama/tingkat atas. Dan sub-tugas dibawahnya dijadikan sub menu yang berkaitan. Tampilan alternatif dapat disesuaikan denga tugas dan peran dari menu.

II.15 Antarmuka

Penggunaan komputer sekarang tidak lagi hanya terjadi pada golongan yang terlibat dalam bidang komputer secara langsung. Komputer telah menjadi salah satu keperluan penting sehari-hari dan digunakan oleh pengguna dengan tahap kemahiran yang berbeda-beda. Pengguna tidak semuanya mahir dalam mengoperasikan komputer. Oleh sebab itu, antarmuka pengguna perlu dibentuk sedemikian rupa sehingga lebih mudah dimengerti dan digunakan oleh pengguna.

Antarmuka merupakan bagian dari komputer dan perangkat lunaknya itu sendiri yang dapat dilihat, didengar, disentuh dan diajak bicara baik secara langsung maupun melalui proses pemahaman tertentu [**8**]. Suatu antarmuka secara tidak langsung juga menunjukkan fungsi sistem kepada pengguna. Antarmuka merupakan gabungan dari elemen-elemensuatu sistem, elemen-elemen dari pengguna, dan juga komunikasi atau interaksi di antara keduanya. Pengguna hanya boleh berinteraksi dengan produk melalui antarmuka pengguna [**8**].

Antarmuka yang baik adalah antarmuka yang tidak disadari dan yang membuat pengguna untuk fokus kepada informasi dan pekerjaannya tanpa perlu mengetahui mekasnisme untuk menyajikan informasi dan melakukan pekerjaan tersebut [**8**]. Sebuah sistem antarmuka pengguna memiliki piranti masukan (seperti *mouse*, *keyboard*, dan media masukan lainnya), piranti keluaran (seperti monitor, suara, dan *printer*), masukan dari pengguna (seperti garis, gerakan mouse, dan sentuhan keyboard) dan hasil yang dikeluarkan oleh komputer (seperti grafik, bunyi, dan tulisan).