# BAB II

# LANDASAN TEORI

## 2.1 Artificial Intelligence

*Artificial Intelligence* merupakan sub bidang pengetahuan komputer yang khusus ditunjukan untuk membuat perangkat lunak dan perangkat keras yang sepenuhnya bisa menirukan beberapa fungsi otak manusia. Dengan demikian diharapkan komputer dapat membantu manusia dalam memecahkan berbagai masalah yang lebih rumit. Komputer yang ada sekarang ini telah berhasil melakukan semua kegiatan pikiran manusia yang paling sederhana dengan baik. Komputer dapat melakukan penyimpanan dan pemanggilan fungsi-fungsi. Komputer dapat melebihi kemampuan manusia terutama dalan hal kecepatan dan penyimpanan data yang lebih besar serta dengan mudah dapat mengambilnya kembali.

### 2.1.1 Arti Intelegensi

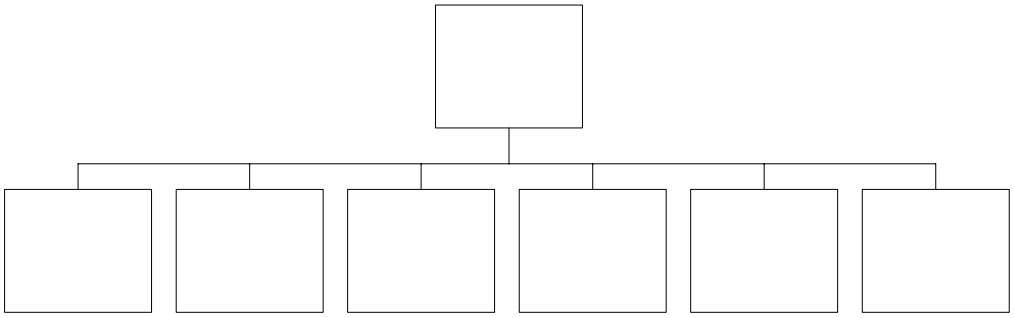
*Intelligence* atau intelegensi adalah seseorang yang pandai melaksanakan pengetahuan yang dimilikinya. Dengan pengertian tersebut maka dapat ditarik suatu kesimpulan bahwa walaupun seseorang memiliki banyak pengetahuan, tetapi jika dia tidak melaksanakannya dalam praktek maka tidak dapat digolongkan kedalam intelegensia. Dengan kata lain, intelegensia adalah kemampuan manusia untuk memperolah pengetahuan dan pandai melaksanakan dalam prakteknya.

Pada batas-batas tertentu *artificial intelligence* memungkinkan komputer untuk dapat menerima pengetahuan melalui masukan dari manusia dan menggunakan pengetahuannya itu untuk melakukan proses penalaran dan berfikir secara manusia untuk dapat memecahkan berbagai masalah.

### 2.1.2 Aplikasi Artificial Intelligence

Aplikasi *artificial intelligence* dapat diterapkan pada komputer, komputer mikro atau komputer pribadi.

Beberapa aplikasi *articial intelligence* diantaranya adalah *General Problem Solving*, *Expert System*, *Natural Language Processing*, *Vision*, *Robotics* dan *Education*.



**Gambar 2.1 Daerah Utama Aplikasi Artificial Intelligence**

1. General Problem Solving

   Pemecahan masalah umum adalah suatu tahapan proses langkah-langkah berurutan untuk mencapai suatu tujuan, dengan cara melacak dan mengkombinasikan berbagai cara atau metode sehingga menghasilkan solusi terbaik ditinjau dari aspek prioritas dan kepentingan.

2. Expert System

   Sistem Pakar adalah suatu program yang bertindak sebagai penasehat atau konsultan pintar dengan mengambil pengetahuan yang disimpan dalam domain tertentu. Seorang pemakai yang belum berpengalaman dalam mendiagnosa penyakit dapat memecahkan suatu masalah yang sulit dan mengambil keputusan dengan tepat hanya dengan bantuan aplikasi sistem pakar ini.

3. Natural Language Processing

   *User* dan program aplikasi dapat berinteraksi menggunakan bahasa alami yang dapat dimengerti dan bahasa tersebut diterjemahkan oleh komputer ke bahasa mesin.

4. Vision

   Aplikasi *artificial* intelligence ini dapat mengenali gambar yang diterimanya dari kamera, *scanner* atau alat input lainnya dengan cara mencocokan dan melacak gambar apa yang diterima melalui kamera atau *scanner* sebagai masukan. Umumnya aplikasi ini dilakukan dengan cara membuat kode untuk setiap gambar, sehingga akan memperkecil jumlah data pada memori.

5. Robotics

   *Artificial intellegence* bertindak seolah-olah melakukan pekerjaan fisik yang dikerjakan oleh manusia, bahkan robot dapat melakukan pekerjaan yang tidak dapat dilakukan oleh manusia atau berbahaya bagi manusia. Semua tindakan dirancang dan disusun dari satu urutan atau algoritma tertentu.

Dengan bantuan robot, pekerjaan yang berulang kali dan rumit dapat dilakukan dengan baik dan dengan hasil yang sama.

6. Education

   Aplikasi *artificial intellegence* ini dalam bidang pendidikan yang bertindak sebagai partner mahasiswa dalam mempelajari suatu hal.

## 2.2 Sistem Pakar

Sistem pakar adalah sistem yang dirancang untuk melakukan jenis-jenis aktivitas yang dapat dilakukan para pakar seperti mendiagnosa, menginterprestasi, merangkum, mengaudit atau memberi nasihat. Pekerjaan yang berhubungan dengan sistem semacam ini biasanya suatu tugas dari dunia-dunia bisnis atau teknik ilmu sains atau ilmu pemerintahan.

Program-program sistem pakar biasanya disiapkan untuk beroperasi dalam suatu keadaan yang akan dipandang cerdas yaitu, seolah-olah ada seorang pakar disisi lain dari terminal video.

Suatu bagian karakteristik dari teknik-teknik pemrograman memberi kehandalan pada program-program ini. Sistem pakar biasanya menggunakan penalaran otomatis dan yang disebut juga metode-metode lemah, teknik-teknik ini cukup berbeda dari algoritma-algoritma yang terhubung dengan baik dan prosedur-prosedur matematika yang mudah dari pemrograman yang lebih tradisional.

Berikut ini adalah daftar singkat secara garis besar tentang beberapa kasus yang menggunakan sistem pakar :

1. Untuk mengkonfigurasi perangkat keras dan lunak dalam suatu komputer agar cocok dengan perencanaan oleh tiap konsumen.
2. Untuk menyetujui permintaan peminjaman, mengevaluasi resiko-resiko asuransi dan kesempatan-kesempatan investasi untuk komunitas finansial.
3. Untuk membantu dokter mendiagnosa dan merawat sekelompok penyakit yang berhubungan, seperti infeksi darah atau berbagai jenis kanker.
4. untuk membantu nahkoda menginterpretasikan data hidrofon dari sekelompok mikrofon di dasar laut yang digunakan untuk mengawasi kapal-kapal dalam area tertentu.
5. untuk membantu ahli kimia menemukan urutan reaksi yang sesuai untuk menciptakan molekul-molekul baru.
6. untuk membantu ahli geologi menginterpretasikan data dari ujung alat pengeboran selama mengeboran minyak.

Kebanyakan aplikasi-aplikasi ini dapat diselesaikan dengan cara yang lebih tradisional maupun dengan bantuan sistem pakar, tapi dalam semua kasus ini ada keuntungannya jika menggunakan sistem pakar.

Dalam beberapa kasus, strategi ini membuat program lebih berorientasi pada manusia. Dalam kasus lainnya, hal ini membuat program menghasilkan keputusan yang lebih baik.

Dalam kasus lainya lagi, penggunaan suatu sistem pakar membuat program lebih mudah dipertahankan dan diperbaharui. Ada banyak keuntungan potensial yang akan dibicarakan nanti.

### 2.2.1 Sistem Pakar Merupakan Salah Satu Program Kecerdasan Buatan

Ada tiga gaya pemrograman yang paling penting yaitu pemrograman sistem pakar, pemrograman kecerdasan buatan dan pemrograman prosedural. Pemrograman sistem pakar merupakan bagian dari unit yang lebih besar yang dapat disebut pemrograman kecerdasan buatan. Pemrograman prosedural dan pemrograman kecerdasan buatan cukup berbeda dalam hal apa yang mereka coba kerjakan dan dalam hal bagaimana mereka mencoba mengerjakannya.

Pemrograman prosedural adalah apa yang setiap orang pelajari ketika mulai menggunakan BASIC, PASCAL atau FORTRAN. Program-program kecerdasan buatan biasanya ditulis dalam bahasa seperti LISP atau PROLOG.

Dalam pemrograman tradisional, komputer harus diberitahukan secara mendetail dan tepat apa yang harus dikerjakan dan bagaimaan mengerjakannya. Gaya ini telah berhasil untuk masalah-masalah yang terdefinisikan dengan baik, bisanya ditemukan dalam pemrosesan data atau dalam pekerjaan teknik serta sains.

Pemrograman kecerdasan buatan kadangkala tampaknya telah didefinisikan sejak awal, seperti yang mudah dikerjakan dalam program-program prosedural tradisional, tapi ada elemen-elemen umum dalam kebanyakan program-program kecerdasan buatan.

Sistem pakar berbeda dari program kecerdasan buatan yang lain oleh tujuannya. Tujuan sistem pakar adalah berusaha untuk melakukan suatu aktifitas yang sesuai bagi seorang pakar.

Sistem pakar memiliki 10 karakteristik yang harus dipenuhi dalam perancangannya. Kesepuluh karakteristik sistem pakar tersebut adalah :

1. Mendukung proses pengambilan keputusan yang menitikberatkan kepada *management perception*.
2. Adanya *human interface* dimana pengguna tetap mengontrol proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah-masalah terstruktur, semi terstruktur dan tidak terstruktur.
4. Menggunakan model-model matematis dan statistik yang sesuai.
5. Interaktif yaitu memiliki kapasitas dialog untuk memperolah informasi sesuai dengan kebutuhan.
6. *Output* ditunjukkan untuk semua orang secara umum.
7. Modularitas yaitu memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
8. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan informasi seluruh tingkatan manajemen.
9. *User friendly* dan fleksibel yaitu mudah untuk digunakan dan memberikan keleluasaan pengguna untuk mengembangkan pendekatan-pendekatan baru.
10. Kemampuan sistem beradaptasi secara cepat dimana pengambilan keputusan dapat menghadapi masalah-masalah baru.

Sistem pakar atau *expert system* menggunakan basis pengetahuan (*knowledge base*) sebagai dasar pemikirannya. *Knowladge base* tersebut terdiri dari *heuristik* dan sejumlah rule atau aturan yang tersusun secara sistematis dan spesifik serta relasi antara data dan aturan atau rule dalam pengambilan kesimpulan. *Knowledge base* tersebut disimpan dalam sebuah struktur data pada suatu tempat penyimpanan data.

Pusat pemrosesan sistem pakar dapat disebut dengan *inference engine* yaitu suatu rancangan aplikasi yang berfungsi untuk memberikan pertanyaan dan menerima masukan dari pengguna, kemudian melakukan proses logika sesuai dengan *knowledge base* yang tersedia. Selanjutnya menghasilkan keluaran berupa suatu kesimpulan atau dapat juga berupa keputusan *decision* sebagai hasil akhir konsultasi.

*Knowledge Acquisition Source* berfungsi sebagai penerjemah dari *knowledge base* menjadi sebuah bahasa yang dapat dimengerti oleh pengguna. Bagian ini diperlukan karena *knowledge base* yang disimpan dalam sebuah *database* disimpan dalam format tertentu, sedemikian rupa sehingga pengguna sulit mengartikannya.

*Working memory* adalah sejumlah modul *memory* yang menyimpan informasi sementara dari suatu proses konsultasi. Setiap kali proses baru dijalankan, *memory* tersebut akan di set ke kondisi awal. Dalam menjalani proses, memory tersebut menyimpan informasi *state* dari rule-rule yang dipakai dalam *knowledge base*.

*Explanation facility* menyimpan data-data *historis* dari sebuah proses konsultasi yaitu rule-rule mana saja yang berperan dalam suatu proses pengambilan keputusan. Data-data *historis* tersebut dapat dijadikan bahan evaluasi dari hasil kerja sebuah *knowledge based system*.

Pengguna memasukan *input* dan menerima *output* melalui sebuah *interface* atau tampilan yaitu sebagai sarana penghubung interaksi antara pengguna dengan sistem.

#### 2.2.1.1 Knowledge Base

*Knowledge base* adalah suatu data yang menyimpan informasi data, rule dan relasi antara data dengan rule dalam pengambilan keputusan. *Knowledge base* dapat dikatakan sebagai kumpulan informasi dan pengalaman seorang ahli pada suatu bidang tertentu. *Knowledge base* terdiri dari fakta seperti halnya buku teori dan *heuristik* yang merupakan langkah-langkah seorang ahli dalam mengambil keputusan.

Setiap kasus yang ditemui akan diproses berdasarkan pengalaman yang telah dicatat, setelah itu baru dapat diambil kesimpulan. Apabila terjadi suatu kasus yang belum pernah dialami, maka sistem pakar ini tidak mampu untuk mengambil kesimpulan dari kasus tersebut. Untuk itu diperlukan penambahan data baru yang dapat dijadikan acuan untuk melengkapi *knowledge base* yang sudah ada. Jadi, suatu *knowledge base* dapat dikatakan baik apabila memiliki sejumlah rule yang mampu digunakan dalam setiap kemungkinan kasus dalam ruang lingkup tertentu.

Dalam perancangannya, perancang harus mampu menggali sebanyak mungkin informasi dari narasumbernya yaitu seorang ahli atau pakar dan memindahkannya kedalam *knowledge base*.

Berdasarkan dari *knowledge base* yang dimilikinya sistem pakar dapat digolongkan menjadi 5 tingkatan yaitu :

1. Tingkat 1
    a. Menggunakan *internal knowledge base*,
    b. Dapat melakukan penambahan data dan membuat laporan data yang ada pada *knowledge base*,
    c. Menggunakan metode *backward chaining* atau metode *forward chaining* untuk menyelesaikan kasusnya,
    d. Mampu menampilkan hasil kerja dari *inference engine* melului *user interface*.
2. Tingkat 2
    a. Menggunakan *external knowledge base*,
    b. Mampu memilih metode *backward chaining* atau *forward chaining* yang akan digunakan untuk menyelesaikan kasusnya,
    c. Mampu melakukan perhitungan matematis.
3. Tingkat 3
    a. Mampu melakukan ekspor atau impor suatu *knowledge base*,
    b. Mampu melakukan perhitungan matematis secara dinamis.
4. Tingkat 4

    Pada tingkatan ini sistem pakar harus dapat dioperasikan pada berbagai sistem operasi (*Operating System*).

5. Tingkat 5

   Pada tingkat kelima ini sistem pakar mampu untuk merubah dan menambah isi *knowledge base* secara otomatis dan mempelajari suatu pola baru dari beberapa kasus yang pernah dialaminya.

#### 2.2.1.2 Dependency Diagram

*Dependency diagram* menunjukan hubungan antara faktor-faktor kritis, pertanyaan yang akan diinputkan, *rule*, *values* dan rekomendasi yang dibuat oleh *knowledge based system*.

#### 2.2.1.3 Decision Tables

*Decision table* diperlukan untuk menunjukan interrelasi antara *value* dari hasil proses pengambilan keputusan *intermediate* maupun keputusan final sebuah *knowledge based system*.

Dalam membuat *decison table* dapat dilakukan langkah-langkah sebagai berikut :

1. Merencanakan jumlah baris yang diperlukan dalam tabel dengan cara mendaftar setiap faktor-faktor dan kondisi-kondisi yang terdapat pada *dependency diagram*,
2. Untuk membuat tabel dapat dimulai dengan membuat tabel kosong berisi nama kolom dan nomor baris, setelah itu isikan setiap value pada tabel tersebut,
3. Mereduksi *decision table* tersebut dengan menggabungkan beberapa rule yang memiliki kesamaan kondisi dan hasil kesimpulan akhirnya.

#### 2.2.1.4 IF-THEN Rules

Dari *decision table* yang telah terreduksi, setiap barisnya dapat dikonversikan menjadi *IF-THEN* rule. Setiap baris pada *decision table* tereduksi akan membentuk satu set rule final.

Struktur dan *syntax* penulisan rule adalah sebagai berikut :

RULE label : Label berisikan nama rule tersebut.

IF : Sebagai penanda awal kondisi pada sebuah rule.

THEN : Sebagai penanda awal kesimpulan pada sebuah rule.

ELSE : Sebagai penanda awal alternatif kesimpulan pada sebuah rule, bersifat operasional, jadi bolah tidak ada.

Operator yang dapat digunakan pada *IF-THEN* rule adalah :

AND : Semua kondisi yang dihubungkan oleh operator ini harus bernilai benar agar kondisi keseluruhan rule tersebut bernilai benar, namun bila ada satu kondisi yang bernilai salah maka keseluruhan rule tersebut dianggap bernilai salah.

OR : Bila semua kondisi yang dihubungkan oleh operator ini harus bernilai salah maka kondisi keseluruhan rule tersebut bernilai salah, namun bila ada satu kondisi atau lebih yang bernilai benar maka keseluruhan rule tersebut dianggap bernilai benar.

#### 2.2.1.5 User Interface

*User interface* adalah tampilan yang akan dihadapkan kepada pengguna, jadi melalui *user interface*-lah pengguna dapat berinteraksi dengan sistem yaitu dengan memasukan *input* dan menerima *output*. Karena itu, perancangan *user interface* haruslah memenuhi prinsip *user friendly*, yaitu mudah digunakan oleh *user* dan meminimalisir kemungkinan terjadinya salah interpretasi.

#### 2.2.1.6 Inference Engine

*Inference Engine* bertindak sebagai pengambil kesimpulan dan mengontrol mekanisme dari *expert system*, karena itu *inference engine* merupakan bagian terpenting dari *expert system*, karena sangat berperan penting dalam menentukan efektivitas dan efisiensi *expert system*. *Inference engine* melakukan proses pengambilan kesimpulan berdasarkan suatu kumpulan rule tertentu.

Metode yang paling umum digunakan dalam proses pengambilan keputuan pada *expert system* adalah modus ponen. Dengan teori dasar yang sederhana saja yaitu apabila premisnya benar, maka kesimpulannya benar, tetapi sebaliknya apabila kesimpulannya benar, maka premisnya belum tentu benar.

Sebagai contoh :

*If* seekor binatang melahirkan

*Then* binatang tersebut adalah mamalia.

Tetapi jika dibalik kondisinya maka akan menjadi :

*If* seekor binatang mamalia

*Then* biatang tersebut melahirkan.

Pernyataan kedua maknanya menjadi salah dikarenakan ada binatang mamalia yang bertelur yaitu *platypus*.

#### 2.2.1.7 Metode Backward Chaining

Strategi ini dimulai dari hipotesa atau dari kondisi sasaran menuju fakta sebagai kondisi awal. *Backward chaining* merupakan penalaran secara induktif. Tujuan diterapkan terlebih dahulu kemudian pengujian atas fakta-fakta yang mendukung tujuan itu. Kebanyakan dari permasalahan mempunyai fakta yang lebih banyak dibanding dengan tujuan, sehingga jika digunakan pelacakan kedepan akan banyak sekali kaidah yang dieksekusi dengan sia-sia, karena ternyata terbukti tidak mengarah kepada sasaran. Pelacakan kebelakang lebih cepat dalam melakukan pelacakan karena akan memilih jalan terpendek.

#### 2.2.1.8 Metode Forward Chaining

*Forward chaining* merupakan kebalikan dari *backward chaining* yaitu memulai dari sekelompok data kemudian dibandingkan dengan fakta-fakta sampai menuju ke kesimpulan.

## 2.3 Pencarian

Terdapat beberapa langkah untuk melakukan proses pencarian dengan baik, pertama yang harus dilakukan adalah mendefinisikan ruang masalah. Ruang masalah ini dapat digambarkan sebagai himpunan keadaan (*state*) atau dapat berupa himpunan rute dari keadaan awal (*initial state*) menuju keadaan tujuan (*goal state*).

Langkah ke dua adalah mendefinisikan aturan produksi yang digunakan untuk mengubah suatu *state* ke *state* lainnya. Langkah terakhir adalah memilih metode pencarian yang tepat sehingga dapat menentukan solusi terbaik dengan usaha yang minimal.

### 2.3.1 Ruang Masalah

Masalah utama dalam membangun sistem berbasis AI adalah bagaimana mengkonversikan situasi yang diberikan ke dalam situasi lainnya yang diinginkan dengan menggunakan sekumpulan operasi tertentu. Untuk melakukan hal ini, diperlukan kemampuan untuk merekayasa sistem.

### 2.3.2 Sistem Produksi

Sistem Produksi terdiri dari :

1. Sebuah himpunan aturan, masing-masing terdiri dari sisi kiri yang menentukan kemampuan aplikasi dari aturan tersebut dan sisi kanan yang menggambarkan operasi yang dilakukan jika aturan tersebut dilaksanakan.
2. Satu atau lebih pengetahuan atau basis data yang berisi informasi apapun untuk tujuan tertentu. Beberapa bagian basis data dapat permanen, dan bagian yang lain dapat hanya berupa solusi untuk masalah saat ini. Informasi dalam basis data ini disusun secara tepat.

3. Strategi kontrol yang menspesifikasikan urutan dimana aturan akan dibandingkan dengan basis data dan menspesifikasikan cara pemecahan masalah jika beberapa aturan dapat dilakukan pada waktu yang sama. Syarat-syarat strategi kontrol:
   a. Cause Motion

      Strategi kontrol yang tidak menyebabkan *motion* tidak akan pernah mencapai solusi.
   b. Systematic

      Strategi kontrol lainya yang dapat dipilih adalah dengan mengacak (*random*) aturan-aturan yang dapat diaplikasikan. Strategi ini lebih baik dari yang pertama, karena menyebabkan *motion*. Pada akhirnya strategi tersebut akan mencapai solusi. Tetapi, mungkin kita akan mengunjungi beberapa *state* yang sama selama proses tersebut dan mungkin menggunakan lebih banyak langkah daripada jumlah langkah minimum yang diperlukan. Beberapa strategi kontrol yang sistematik telah diusulkan, yang disebut sebagai metode-metode pencarian.
4. Pengaplikasi aturan (*a rule applier*) untuk mengaplikasikan aturan yang dipilih.

### 2.3.3 Metode- metode Pencarian

Terdapat banyak metode pencarian ysng telah diusulkan. Semua metode yang ada dapat dibedakan ke dalam dua jenis yaitu pecarian buta atau tanpa informasi (*blind* atau *un-informed search*) dan pencarian *heuristik* atau dengan informasi (*heuristic* atau *informed search*). Setiap metode mempunyai karakteristik yang berbeda dengan kelebihan dan kekurangannya masing-masing.

Untuk mengukur performansi metode pencarian, terdapat empat kriteria yang digunakan yaitu:

1. *Completeness*: Apakah metode tersebut menjamin penemuan solusi jika solusi memang ada?
2. *Time Complexity*: Berapa lama waktu yang diperlukan?
3. *Space Complexity*: Berapa banyak memori yang diperlukan?
4. *Optimality*: Apakah metode tersebut menjamin menemukan solusi yang terbaik jika terdapat beberapa solusi berbeda?

#### 2.3.3.1 Blind (Un-informed Search)

Penggunaan istilah *blind* atau buta karena memang tidak ada informasi awal yang digunakan dalam proses pencarian. Terdapat beberapa metode yang tergolong *blind search,* diantaranya:

1. Breadth First Search (BFS)

   Proses pencarian dilakukan pada semua simpul dalam setiap level secara berurutan dari kiri ke kanan. Jika pada satu level belum ditemukan solusi, maka pencarian akan dilanjutkan pada level berikutnya.

Demikian seterusnya sampai ditemukan solusi (jika solusinya memang ada) dan solusi yang ditemukan pasti yang paling baik. Dengan kata lain BFS adalah *complete* dan *optimal*. Tetapi, BFS harus menyimpan semua simpul yang pernah dibangkitkan. Hal ini harus dilakukan agar BFS dapat melakukan penelusuran simpul-simpul sampai di level bawah.

2. Depth First Search (DFS)

   Proses pencarian dilakukan pada satu simpul dalam setiap level dari yang paling kiri. Jika pada level yang terdalam solusi belum ditemukan maka pencarian dilanjutkan pada simpul sebelah kanan dan simpul yang kiri dapat dihapus dari memori. Jika pada level yang paling dalam tidak ditemukan solusi, maka pencarian dilanjutkan pada level sebelumnya. Demikian seterusnya sampai ditemukan solusi. Kelebihan DFS adalah pemakaian memori yang tergolong kecil dibandingkan dengan metode pencarian lainnya.

3. Depth Limited Search (DLS)

   Metode ini berusaha mengatasi kelemahan DFS (tidak *complete*) dengan membatasi kedalaman maksimum dari suatu jalur solusi. Tetapi, sebelum menggunakan DLS, kita harus mengetahui berapa level maksimum dari suatu solusi. Jika batasan kedalaman terlalu kecil, DLS tidak dapat menemukan solusi yang ada. Artinya, DLS dapat menjadi tidak *complete* jika batasan kedalamannya lebih kecil dibandingkan dengan level solusinya.

### 2.3.3.2 Metode Pencarian Heuristik

Kata *heuristic* berasal dari sebuah kata kerja bahasa Yunani, *heuriskein*, yang berarti ’mencari’ atau ’menemukan’. Dalam dunia pemrograman, sebagian orang menggunakan kata heuristik sebagai lawan dari algoritmik, dimana kata heuristik ini diartikan sebagai ’suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditemukan’. Kata heuristik diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi.

Metode-metode yang termasuk kedalam teknik pencarian yang berdasarkan pada fungsi heuristik, diantaranya :

1. Generate and Tes (Bangkitkan dan Uji)

   Metode *Generate and Test* (GT) adalah metode yang paling sederhana dalam teknik pencarian heuristik. Jika pembangkitan sebuah solusi yang mungkin (*a possible solution*) dikerjakan secara sistematis, maka prosedur ini menjamin akan menemukan solusinya. Tetapi, jika ruang masalahnya sangat luas, mungkin memerlukan waktu yang sangat lama. Algoritma GT menggunakan prosedur *Depth First Search* (DFS) karena solusi harus dibangkitkan secara lengkap sebelum dilakukan *test*. Algoritma ini berbentuk sistematis, pencarian sederhana yang mendalam dari suatu ruang permasalahan. GT juga dapat dilakukan dengan pembangkitan solusi secara acak, tetapi tidak ada jaminan solusinya akan ditemukan.

2. Hill Climbing (Pendakian Bukit)

   *Hill Climbing* (HC) berbeda dengan GT, yaitu pada umpan balik prosedur tes untuk membantu pembangkit menemukan solusi yang bisa langsung dihilangkan dalam ruang pencarian. Dalam GT, respon dari fungsi pengujian hanya berupa ’ya’ atau ’tidak’. Tetapi jika pengujiannya ditambahkan dengan fungsi yang menghitung biaya estimasi antara *state* sekarang dengan *state* tujuan, maka prosedur pembangkit dapat memanfaatkan nilai ini. HC sering digunakan jika terdapat suatu fungsi heuristik yang terbaik untuk mengevaluasi *state*.

3. Best First Search (Pencarian Terbaik Lenih Dulu)

   Metode *Best First Search* memilih simpul baru yang memiliki biaya terkecil diantara semua *left nodes* (simpul-smpul pada level terdalam) yang pernah dibangkitkan. Dengan demikian algoritma *Best First Search* dalam proses pencariannya melakukan penghitungan yang lebih banyak dibandingkan dengan metode pencarian heuristik lainnya.

## 2.4 Konsep Dasar Perancangan Sistem

Perancangan sistem secara umum adalah suatu tahap dimana didalamnya terdapat identifikasi komponen-konponen sistem yang akan dirancang secara rinci yang bertujuan untuk memberikan gambaran kepada pengguna atau *user* mengenai sistem yang baru. Sedangkan desain secara terinci dimaksudkan untuk pembuat program komputer dan ahli teknik lainya yang akan mengimplementasikan sistem.

Penggambaran dan rancangan model sistem secara logika dapat dibuat dalam bentuk Diagram Konteks dan Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD).

### 2.4.1 Diagram Konteks

Diagram Konteks adalah diagram tingkat tinggi dari Diagram Alir Data yang merupakan gambaran global dari sistem yang menggambarkan aliran-aliran data kedalam maupun keluar suatu sistem dan merupakan alat yang digunakan untuk melihat batasan antara sistem dengan *eksternal entity*.

### 2.4.2 Diagram Alir Data

Diagram Alir Data atau *Data Flow Diagram* (DFD) adalah suatu model yang menjelaskan arus data mulai dari pemasukan sampai dengan keluaran data. Tingkatan DFD dimulai dari diagram konteks yang menjelaskan secara umum suatu sistem atau batasan sistem aplikasi yang akan dikembangkan. Kemudian DFD dikembangkan menjadi DFD tingkat 0 atau level 0 dan kemudian DFD level 0 dikembangkan lagi menjadi level 1 dan selanjutnya sampai sistem tersebut tergambarkan secara rinci menjadi tingkatan-tingkatan lebih rendah lagi.

DFD merupakan penurunan atau penjabaran dari diagram konteks. Dalam pembuatan DFD harus mengacu pada ketentuan sebagai berikut :

1. Setiap penurunan level yang lebih rendah harus mempresentasikan proses tersebut dalam spesifikasi proses yang jelas.
2. Penurunan dilakukan apabila memang diperlukan.

3. Tidak semua bagian dari sistem harus ditunjukan dengan jumlah level yang sama.

## 2.5 Konsep Basis Data

Basis data adalah kumpulan *file* yang saling berinteraksi, relasi tersebut bisa ditunjukan dengan kunci dari tiap *file* yang ada. Satu basis data menunjukan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi.

Terdapat dua buah teknik perancangan basis data, yaitu dengan membuat *Entity Relationship Diagram* atau dengan menerapkan normalisasi terhadap struktur yang telah diketahui.

### 2.5.1 Normalisasi

Normalisasi adalah sebuah teknik untuk mengoptimalkan rancangan basis data relasional dan membebaskan rancangan tersebut dari keganjilan dan persoalan yang potensial. Normalisasi juga dapat diterapkan ke model basis data lainnya. Secara sederhana, normalisasi juga dapat diterapkan ke model basis data dalam tabel yang lebih kecil sampai tiap atribut dalam tiap tabel hanya bergantung pada (beberapa) kunci dalam tabel tersebut. Adapun tujuan dari normalisasi ini adalah :

1. Meminimumkan duplikasi data,
2. Menyediakan fleksibilitas yang diperlukan untuk kebutuhan fungsional yang berbeda,
3. Memungkinkan suatu model untuk digambarkan dalam suatu basis data,

4. Berguna untuk mengstrukturkan data dalam cara-cara tertentu untuk membantu mengurangi atau mencegah munculnya masalah yang berhubungan dengan pengolahan data.

Rancangan basis data yang buruk atau yang tidak dinormalisai akan menyebabkan persoalan selama basis data tersebut di*instal.* Persoalan yang ringan, basis data menjadi tidak efisien untuk dijalankan dan sulit untuk dipelihara. Sedangkan persoalan yang terberat adalah setelah basis data di*install*, pemakai menemukan bahwa basis data tersebut tidak menghasilkan apa yang dibutuhkan atau memberikan hasil yang tidak akurat.

### 2.5.2 Kamus Data

Kamus data atau *data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem. Kamus data berisikan beberapa tabel yang memuat nama dari data, tipe dari data yang akan kita gunakan didalam menyusun program. Kamus data digunakan untuk menghindari duplikasi elemen-elemen dan menghindari konflik antara elemen-elemen tersebut. Dengan kamus data, dapat dilihat elemen-elemen yang terdapat dalam sebuah sistem.

## 2.6 Tipe Jaringan

Dalam jaringan komputer terdapat dua jenis jaringan, yaitu :

### 2.6.1 Model *Peer to Peer*

Pada model jaringan ini terdapat hanya dua komputer yang saling terhubung. Masing-masing komputer mempunyai kedudukan yang setara atau setingkat sehingga komputer manapun dapat memulai (*initiate*) dan mengendalikan hubungan dalam jaringan. Model Jaringan ini merupakan jenis yang paling sederhana.




**Gambar 2.2 Model Jaringan *Peer to peer***

### 2.6.2 Model *Client server*

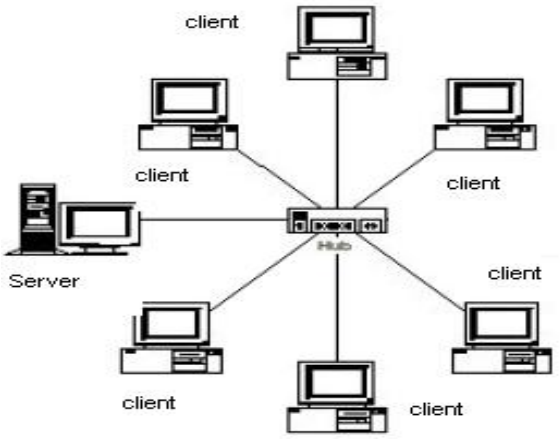
Konsep Jaringan *client-server* ini membedakan dengan jelas kedudukan suatu komputer, mana yang dapat memberikan layanan jaringan (*server*) dan mana yang hanya meminta layanan (*client*). Jika suatu komputer di*install* sebagai *server* yang memberikan segala sumber daya (*resource*), contohnya : *printer*, *modem* kepada komputer lain (*client)* yang terhubung ke jaringan. Untuk dapat saling berkomunikasi antara *server* dan *client*, maka digunakan suatu aplikasi jaringan yang dinamakan *client-server program*, dimana *server* menggunakan aplikasi jaringan yang disebut *server program*, sementara *client* menggunakan *client program* untuk dapat berkomunikasi dengan *server program* pada *server*.

Beberapa sifat dari *client-server* diantaranya sebagai berikut :

b. *Server* dan *Client* berada pada posisi serta proses yang berbeda.
c. *Server* dan *Client* dapat dijalankan pada mesin yang sama atau berbeda.
d. Setiap proses dapat menyembunyikan informasi tersendiri.
e. Komunikasi yang dilakukan bisa bersifat protokol *peer to peer*.

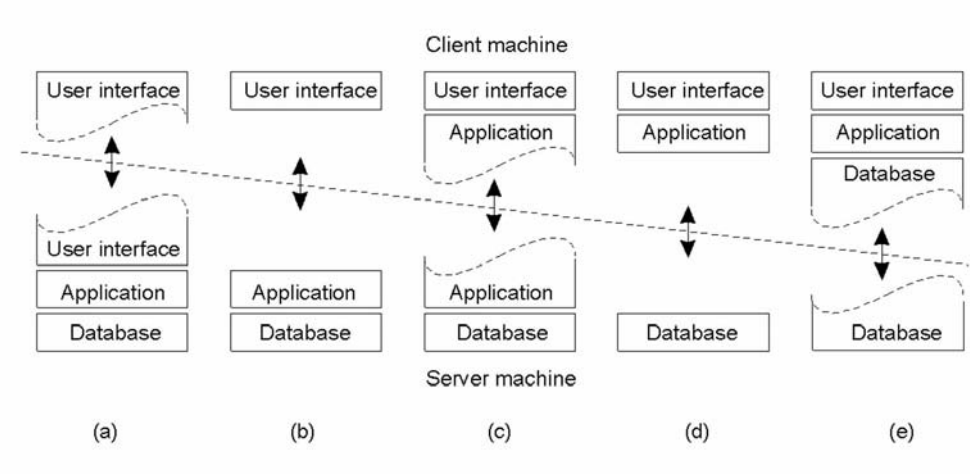
Koneksitas *client ke server* :

a. *Server* harus terlebih dahulu dijalankan (*running*).
b. *Server* harus menentukan *port* untuk koneksi yang masuk.
c. *Client* harus mengetahui nama *host* dari *server* beserta *port* yang digunakan untuk koneksi.
d. *Client* mengirim pesan ke *server* untuk melakukan suatu pekerjaan (*task*).




**Gambar 2.3 Model Jaringan *Client Server***

Dalam *client-server program* terdapat berbagai alternatif pemodelan aplikasinya, diantaranya dapat dilihat pada gambar dibawah ini.




**Gambar 2.4 Model Alternatif Client – Server**

## 2.7 Tinjauan Perangkat Lunak

Kebutuhan perangkat lunak adalah suatu kondisi atau keadaan yang harus dipenuhi atau dimiliki oleh suatu sistem, atau suatu kondisi keadaan yang diperlukan oleh pemakai untuk mencapai tujuan.

### 2.7.1 Borland Delphi 7

Borland delphi adalah suatu aplikasi pengembangan perangkat lunak yang berjalan dilingkungan Windows. Borland Delphi menggunakan bahasa pemrograman Pascal. Borland Delphi sendiri merupakan hasil pengembangan bahasa Pascal yang bersifat visual, yang dikembangkan oleh Borland International. Aplikasi yang dapat dikembangkan dengan Borland Delphi berupa game, database, multimedia dan sebagainya.

Setelah muncul versi pertama dari Borland Delphi, setiap release Borland Dephi selalu disertai fitur baru dan perbaikan dari versi sebelumnya.

#### 2.7.1.1 Fitur pada Dephi 7.0

Fitur baru dan perbaikan yang ada pada Borland Delphi 7 ini adalah :

1. IDE (Interface Development Environtment)

   Lingkungan pengembang aplikasi (IDE) Borland Delphi 7 telah mengalami perubahan dari versi sebelumnya. Diantaranya, terdapat *Compiler Message*, perubahan pada *Component Pallet*, *Code Insight* dan *Debugger*.

2. Web

   Borland Delphi 7 menyediakan Intraweb buatan AtoZed Software, yang dapat digunakan untuk membuat aplikasi web server dengan sarana standar. Borland Delphi 7 juga mendukung pada Apache 2. Borland juga menghilangkan Win-CGI sebagai target aplikasi Web Server dan Web Service. Fasilitas untuk Web Server juga mengalami perbaikan-perbaikan.

3. COM

   Sekarang ini dengan Delphi 7.0 dapat membuat CoClass Wrapper bagi pengembang .NET dengan cara menggunakan kotak dialog Import Type Library. Dengan adanya resulting wrapper, maka fitur interoperabilitas dari Microsoft’s .NET Framework dapat digunakan.

4. Database

   Pada Delphi 7.0 driver dbExpress telah diupdate bagi Informaix SE, Oracel 9i, DB2 7.0, InterBase 6.5 dan MySQL 3.2.3.4.9. Driver baru bagi MSSQL 2000 juga tersedia. Disamping itu, beberapa hal baru dan perubahan juga dilakukan pada komponen database.

   Borland juga telah membuang SQL Links. Borland merekomendasikan pemakaian dbExpress bagi database SQL Server yang diakses di Delphi.

5. Component Library

   Jika ditelusuri komponen library Delphi 7.0, maka akan ditemukan komponen baru, unit baru, komponen yang berubah, komponen yang hilang dan komponen yang mendukung bagi tema Windows XP.

6. Runtime Library

   Beberapa perubahan di Runtime Library antara lain ialah perubahan pada unit Classses, Math, StdConv, StrUtils, SysUtils, VarCmplx dan Variants.

7. Compiler

   Kompiler Delphi dcc32 sekarang ini support terhadap tiga warning kompiler tambahan, yaitu Unsafe_Type, Unsafe_Code dan Unsafe_Case. Warnings tersebut defaultnya adalah *disabled*, tetapi dapat di-*enabled*. Fitur ini sangat membantu kita ketika akan mem-port kode ke lingkungan eksekusi terkendali di platform Microsoft’s .NET.

8. Model Maker

   Sarana baru yang disebut ModelMaker dapat membantu memudahkan proses desain, kontruksi dan pengolahan class dan interface. ModelMaker juga memiliki sarana untuk pembuatan diagramUML-style, yang dapat dipakai untuk membuat dan memodifikasi *source code project*.

## 2.8 Olahraga

Olahraga kian meluas dan populer di Indonesia. Secara umum hal ini dapat dikemukakan, umpamanya berdasarkan jumlah cabang yang pernah dipertandingkan dalam PON XI/1985 yaitu 43 cabang olahaga. Dalam konteks PON banyak orang tak sependapat, misalnya mengapa *aeromodeling* atau catur dimasukan dalam kategori cabang olahraga. Karena banyaknya kritik, akhirnya pada PON XII/1989 cabang yang dipertandingkan menciut menjadi 33 cabang olahraga. Hingga kini memang masih banyak silang pendapat tentang apa yang disebut olahraga, kesulitan yang dihadapi terutama di kalangan ilmuan yaitu kian banyak cabang olahraga yang muncul dalam kehidupan modern dan nyata diakui sebagai olahraga oleh masyarakat.

Bahkan cabang olahraga itu juga secara berkala dipertandingkan pada tingkat nasional dan internasional.

Berbagai rumusan yang berbeda tentang definisi olehraga terutama berpangkal dari sudut pandang yang berlainan. Smith (1971) mengemukakan bahwa olahraga merupakan perluasan dari bermain. Karena itu olahraga berlandaskan pada bermain dan nilai inti dalam kegiatan itu.

Tiga ciri pokok olahraga yang dianggap mematikan semangat bermain adalah :

1. Penekanan kemenangan,
2. Pengutamaan teknik dan keterampilan yang baik,
3. Kehadiran penonton yang berlebihan.

Matveyev (1981) mengatakan, olahraga merupakan satu kegiatan olahraga yang energik dan dalam kegiatan itu atlet memperagakan kemampuan geraknya dan kemauannya semaksimal mungkin. Dari dua definisi olahraga diatas maka dapat dikatakan olahraga memerlukan peragaan ketangkasan fisik yang terungkap dalam keterampilan, kesegaran jasmani atau kombinasi kedua hal itu. Disamping mengandung ciri bermain, olahraga bersifat kompetitif, karena itu dilibatkan keterampilan, strategi dan pemanfaatan peluang.

### 2.8.1 Ilmu Keolahragaan

Kemajuan yang dicapai dalam olahraga pada umumnya terutama berkat dukungan ilmu pengetahuan dan teknologi (IPTEK). Produk penelitian dalam alat-alat olahraga yang lebih canggih,

pengembangan alat ukur dan penganalisis yang lebih cermat, metode yang lebih berhasil dan penemuan dalam berbagai bidang disiplin ilmu saling terpadu.

Apa yang dimaksud dengan ilmu keolahragaan. Masih kuat dalam pandangan yang beredar di kalangan masyarakat awam yakni pelaksanaan olahraga yang dapat dilakukan tanpa dukungan ilmu pengetahuan dan teknologi.

Tapi jika dikaitkan dengan kenyataan sekarang, ternyata tak memadai untuk memecahkan masalah. Dukungan IPTEK merupakan syarat mutlak.

Para ahli telah merintis usaha untuk mengidentifikasi struktur ilmu keolahragaan, disini digunakan istilah struktur, karena ilmu keolahragaan bukanlah ilmu yang berdiri sendiri, tapi sebagian lintas disiplin, artinya, terdapat seperangkat disiplin ilmu yang saling mendukung. Selain ada keterpaduan antara disiplin ilmu, juga ada pengembangan disiplin masing-masing.

Susunan batang ilmu keolahragaan dapat dijelaskan berdasarkan tiga kemungkinan pendekatan yang menekankan pada :

1. Isi,
2. Tema,
3. Bidang teori.

Tugas utama ilmu keolahragaan adalah :

1. Mengidentifikasi status kesehatan para olahragawan dalam berolahraga,
2. Menggambarkan perkembangan fisik para olahragawan,
3. mengidentifikasi status fungsi organ tubuh dan kemampuan para olahragawan.

Penerapan ilmu pengetahuan dan teknologi bertujuan untuk memaksimalkan hasil yang dicapai. Asas penghematan, manfaat dan kegunaan praktis akan sangat menonjol dalam olahraga. Persoalan berikutnya adalah sampai batas mana dapat leluasa menerapkan IPTEK dalam olahraga.

### 2.8.2 Pemanasan Bagi Olahragawan (Warm Up)

Kebanyakan atlet melakukan pemanasan (*warm up*) sebelum melakukan latihan-latihan. Jika ditanya mengapa melakukanya, bisanya jawabannya adalah agar otot-otot tidak kaku. Jarang sekali yang mengetahui kenaikan suhu tubuh yang disebabkan melakukan pemanasan sangat bermanfaat bagi penampilan fisik.

Pemanasan sangat penting artinya bagi penampilan fisik yang membutuhkan kemampuan puncak dalam waktu singkat, misalnya lari jarak pendek, angkat beban atau lompat tinggi. Efek kenaikan suhu tubuh pada pemanasan lebih penting artinya daripada bertambah cepatnya pernapasan dan sistem peredaran darah serta juga lebih penting daripada kenaikan kadar hormon.

Dalam penelitian, ternyata disosiasi *hemoglobin* lebih cepat dan lebih sempurna jika suhu tubuh naik. Demikian juga proses oksidasi akan lebih meningkat pada otot yang aktif bekerja. Keluarnya oksigen dari *mioglobin* yang merupakan sumber tambahan oksigen yang terdapat pada otot-otot kerangka dan otot jantung juga mengalami peningkatan pada suhu tubuh yang lebih tinggi.

Kenaikan suhu juga akan mengurangi kekentalan *intern protoplasma* otot, sehingga dapat memperbaiki efisiensi mekanis otot-otot yang berkontraksi. Otot-otot dapat berkontraksi lebih cepat dan lebih keras jika suhu otot sedikit lebih tinggi daripada suhu tubuh, sebaliknya jika suhu otot lebih rendah daripada suhu tubuh yang normal maka *viskositas* (kekentalan) plasma otot akan bertambah, akibatnya otot tersebut akan menjadi lamban dan kurang kuat berkontraksi.

Bertambahnya suhu tubuh juga mempercepat jalannya impuls saraf (rangkaian saraf) dan menambah kepekaan reseptor-reseptor saraf, sehingga mengakibatkan gerak seseorang menjadi lebih cepat,

ini disebabkan oleh koordinasi yang bertambah baik, karena itu perbaikan fungsi sistem saraf yang disebabkan pemanasan sangat bermanfaat bagi atlet-atlet bulu tangkis, tenis, bola basket dan olahraga lainya yang membutuhkan gerak-gerak kompleks bagian-bagian tubuh.

Kenaikan suhu tubuh ini juga merupakan rangsangan pada pembuluh darah untuk melebar, sehingga menambah banyaknya aliran darah dari otot-otot. Mekanisme ini dapat memperlancar pengiriman zat-zat yang dibutuhkan oleh otot-otot yang bergerak pada waktu melakukan olahraga tadi dan dapat mengurangi penimbunan asam laktat.

Pemanasan yang cukup dapat mencegah terjadinya cedera pada otot, tendo, ligmen dan jaringan pengikat lainnya. Elastisitas otot tergantung pada cukup atau tidaknya darah yang berada dalam otot tadi. Jadi pada otot yang dingin jumlah darah akan berkurang, sehingga lebih mudah mengalami cedera daripada otot yang panas. Kelenturan tendo dan ligamen juga sangat dipengaruhi oleh suhu.

Terdapat tiga macam tipe dasar pemanasan yang dapat digunakan dalam persiapan olahraga, masing-masing cukup efektif karena dapat menaikan suhu rectal dan otot. Ketiga tipe pemanasan tersebut adalah :

1. Pasif
2. Nonspesifik (umum)
3. Spesifik

Pada tipe pasif, pemanasan dilakukan dengan bantuan dari luar tubuh misalnya mandi air panas atau dengan bantal panas. Meskipun cara ini tidak praktis asalkan dapat menaikan suhu tubuh, maka penampilan atlet dapat menjadi lebih baik.

Pada tipe nonspesifik, suhu tubuh dinaikkan dengan gerakan-gerakan kelompok otot besar yang aktif, misalnya senam yang tidak secara langsung berhubungan dengan komponen-komponen otot dan saraf yang akan digunakan pada waktu olahraga. Keuntungan tipe ini dibandingkan dengan tipe pasif adalah kenaikan suhu otot menjadi lebih efektif. Oleh karena itu dari segi olahraga tipe ini lebih menguntungkan.

Pada tipe spesifik, pemanasan dikhususkan pada bagian-bagian otot dan saraf tubuh yang akan dipergunakan pada olahraga. Misalnya seorang atlet angkat besi akan mempersiapkan kemampuan maksimalnya pada pertandingan *benchpress*, maka ia melakukan pemanasan dengan angkatan *benchpress* dengan beban dari yang paling ringan dan secara bertahap beban dinaikan sampai mendekati batas maksimalnya (kurang lebih 70 - 80 % dari beban maksimal).

Pemanasan yang spesifik ini tidak hanya menaikan suhu tubuh atau bagian tubuh yang akan dipertandingkan pada waktu melakukan olahraga, tetapi juga memberikan kesempatan pada atlet tersebut untuk mengadakan percobaan sebelum pertandingan. Oleh karena itu, pemanasan spesifik ini merupakan pemanasan yang paling disenangi dan banyak dilakukan.

### 2.8.3 Cedera Pada Olahragawan

Umumnya cedera olahraga terjadi akibat ketidakseimbangan antara beban kerja dengan kemampuan jaringan yang terkait.

Cedera olahraga dapat diklasifikasikan dalam 2 bagian yaitu:

1. *Eksternal* atau *Exposed* yaitu terjadinya robekan kulit.
2. *Internal* atau *Unexposed* yaitu tidak terdapat robekan kulit.

selain itu cedera olahraga dapat bersifat :

1. Akut : Terjadi sangat tiba-tiba, biasanya berlangsung singkat dan lebih berpengaruh pada atlet.
2. Khronik : Terjadi akibat truma yang berlangsung lama serta respon pengobatan lamban.

Kerusakan struktur otot relatif sering terjadi. Perlu dibedakan antara robek otot lengkap dan tidak tengkap. Cedera jaringan otot kebanyakan terjadi akibat pembebanan secara tiba-tiba dan koordinasi gerakan yang kurang baik. Selain itu dapat pula terjadi karena kelelahan dan pemanasan yang tidak sempurna. Otot yang mengalamai memar cenderung menjadi *myosistis ossificans* akibat pengobatan fisioterapis yang salah.

Cedera pada tendo terjadi akibat kekuatan yang datang secara tiba-tiba, terutama pada tendo yang telah mengalami kerusakan sebelumnya. Kerusakan disebabkan oleh *micro-fissure* pada tendo atau penyuntikan *kortikosteroid*.

### 2.8.4 Definisi Dan Aspek-aspek Pelatihan

Definisi latihan adalah suatu proses berlatih yang sistematis yang dilakukan secara berulang-ulang, dan kian hari jumlah beban kian bertambah.

Tujuan utama pelatihan olahraga adalah untuk meningkatkan keterampilan atau prestasi semaksimal mungkin. Untuk mencapai tujuan itu ada empat aspek latihan yang perlu dilatih secara seksama yaitu :

1. Latihan Fisik,
2. Latihan Teknik,
3. Latihan Taktik,
4. Latihan Mental.

#### 2.8.4.1 Intensitas Latihan

Perubahan fisiologis dan psikologis hanyalah mungkin terjadi apabila latihan dilakukan secara intensif. Latihan intensif adalah proses latihan haruslah kian berat dengan cara menambah beban kerjanya, jumlah repetisi gerakan serta kadar intensitas pengulangan gerak. Latihan yang ringan tidak akan dapat merangsang perubahan dalam fungsi organ tubuh dalam hal yang bersifat kejiwaan

Orang sering bertanya bagaimanakah menentukan kadar intensitas latihan. Bagaimanakah menentukan apakah suatu latihan intensif atau tidak. Demikian pula berapa lama latihan yang intensif itu harus dilaksanakan sebagai tolak ukur menentukan kadar intensitas latihan, khususnya untuk perkembangan daya tahan *kardiovaskuler*, untuk kasus seperti ini dapat diterapkan teori Katch dan McArdle (1983) sebagai berikut :

1. Mula-mula kita hitung frekuensi Denyut Nadi Maksimal (DNM) yang rumusnya adalah

   Deyut Nadi Maksimal = 220 - UMUR

   Jadi DNM seorang atlet yang berusia 20 tahun adalah, 220 – 20 = 200 denyut nadi per menit.

2. Selanjutnya kita ukur takaran intensitas latihannya. Bagi seorang atlet olahraga prestasi, takaran intensitas latihannya sebaiknya adalah

   80 % hingga 90 % dari DNM

   Jadi takaran intensitas latihan bagi atlet yang berumur 20 tahun adalah 80 – 90 % dari 200 = 160 – 180 denyut nadi per menit, untuk gambaran, jika anda melakukan kegiatan *warming-up* berupa jogging beberapa ratus meter lalu dilajutkan dengan latihan senam, maka denyut nadi anda adalah kira-kira 110 – 220 denyut nadi per menit. Jadi dapat anda bayangkan berapa intensif latihan harus dilakukan untuk mencatat 160 – 180 denyut nadi per menit.

   Jika atlet mampu mencapai takaran intensitas tersebut, maka dia dapat dikatakan telah berlatih dalam zona latihan (*training zone*).

3. Intensitas latihan juga ditentukan oleh lamanya berlatih dalam zona latihan. Seorang atlit harus berlatih dalam zona latihan selama 45 hingga 120 menit untuk benar-benar berlatih intensif.

Bagi orang yang bukan atlet yang berolahraga sekedar untuk menjaga kesehatan atau memelihara kondisi fisiknya, intensitas latihannya tidak perlu seberat untuk atlet. Patokannya adalah :

| |
|---|
| 70 % hingga 85 % dari DNM |

Sebagai contoh, takaran intensitas latihan untuk seorang yang berusia 40 tahun adalah 70 – 85 % dari (220 – 40) = 126 – 153 denyut per menit. Lama berlatih pun tidak perlu seberat atlet, cukup 20 hingga 30 menit saja berlatih dalam zona latihan.

Jadi jelas bahwa agar dapat disebut berlatih intensif, orang harus memenuhi persyaratan seperti pada langkah 1, 2 dan 3.

Latihan akan kurang manfaatnya jika tidak memenuhi persyaratan tadi.

### 2.8.4.2 Kualitas Latihan

Berlatih secara intensif belumlah cukup menjamin tercapainya peningkatan prestasi, terutama jika latihannya tidak bermutu. Orang bisa saja berlatih keras, intensif, sampai habis tenaga, tetapi dikarenakan latihannya tidak bermutu maka peningkatan prestasinya pun tak terjadi.

Adapun rumus untuk mencari Index Massa Berat Tubuh adalah

$$IMT = \frac{\text{Berat Badan (Kg)}}{\text{Tinggi Badan}^2\ \text{(m)}}$$

Keterangan :

| | |
|---|---|
| Berat Badan Kurang | Obesitas I |
| IMT < 18,5 | IMT = 30 – 34,9 |
| Berat Badan Normal | Obesitas II |
| IMT = 18,5 – 24,9 | IMT = 35 – 39,9 |
| Berat Badan Lebih | Sangat Obesitas |
| IMT = 25 – 29,9 | IMT > 39,9 |

Keterangan Frekuensi Latihan

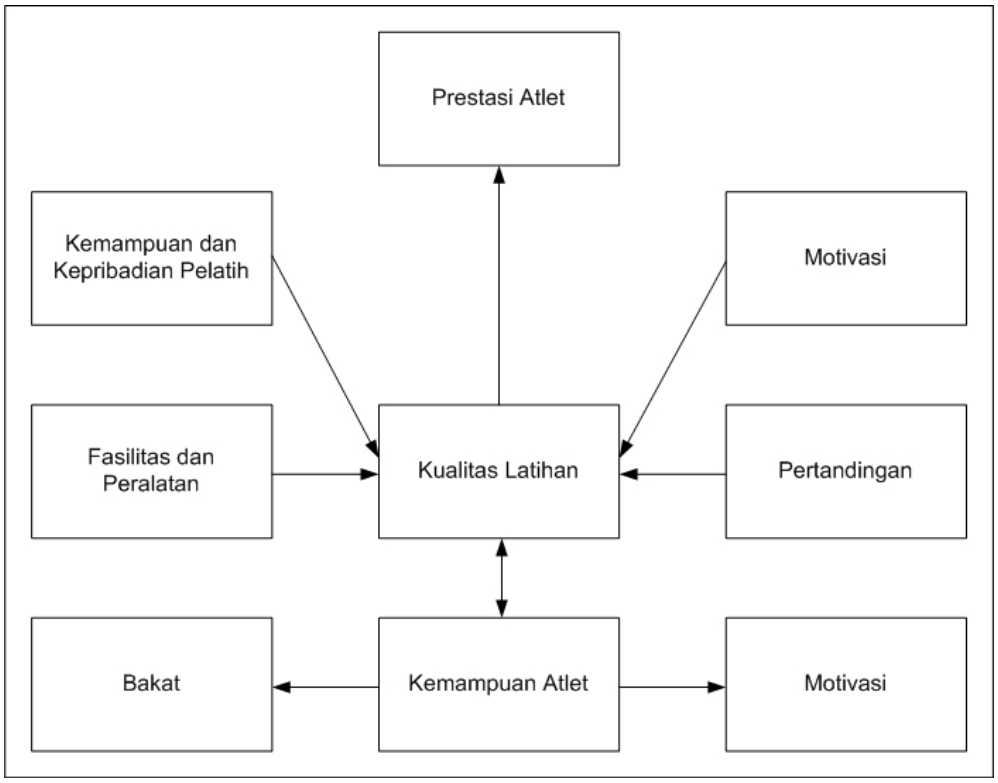
1. 3 kali seminggu apabila berat badan kurang.
2. 4 kali seminggu apabila berat badan normal.
3. 5 kali seminggu apabila berat badan lebih dan obesitas I
4. 6 kali seminggu apabila berat badan obesitas II dan sangat obesitas.

Apakah makna atau maksud latihan yang bermutu itu, beberapa pertanda latihan yang bermutu adalah sebagai berikut :

1. Latihan atau *dril* yang diberikan oleh pelatih adalah benar-benar bermanfaat dan sesuai dengan kebutuhan atlet.
2. Koreksi yang tepat dan konstruktif selalu diberikan manakala atlet melakukan kesalahan-kesalahan.
3. Pengawasan terhadap setiap detil gerakan dilakukan secara teliti.
4. Setiap kesalahan gerakan segera diperbaiki.

Meskipun kurang intensif, latihan yang bermutu seringkali lebih bermanfaat daripada latihan intensif tetapi tak bermutu. Bermutu tidaknya latihan sangat tergantung pada kepandaian dan kejelian pelatih dalam merancang program latihan.

Kecuali kemampuan melatih, banyak faktor yang ikut mendukung dan menentukan mutu latihan, misalnya hasil-hasil penelitian di bidang pelatihan, fasilitas dan perlengkapan latihan, hasil evaluasi dan analisis pertandingan yang lalu, kemampuan dan bakat atlet.




**Gambar 2.5 Kualitas Latihan dan Faktor-faktor Pendukung**

Kekeliruan pelatih atau atlet adalah lebih menekankan pada lamanya latihan ketimbang pada mutu dan penambahan beban latihannya. Karena itu sebaiknya pendek dan berisi serta padat dengan kegiatan-kegiatan yang bermanfaat. Jika latihan berlangsung terlalu lama dan terlalu melelahkan maka bahayanya adalah atlet akan memandang setiap latihan sebagai siksaan dan hari-hari latihan berikutnya akan ditatapnya dengan perasaan enggan.