BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Instansi

2.1.1. Sejarah Antam

. Antam merupakan sebuah perseroan terbatas di Indonesia dan termasuk perusahaan negara yang terintegrasi secara vertikal untuk melakukan semua tahapan proses penambangan dari eksplorasi, pertambangan, peleburan, dan pemurnian melalui bagian pemasaran. Produk utama Antam adalah feronikel, bijih nikel, emas, perak dan bauksit. Antam 35% dipegang oleh publik, yang sebagian besar adalah dengan lembaga asing, yang telah mengadakan Antam untuk beberapa tahun. Terdaftar di Indonesia dan Bursa Efek Australia, Antam adalah salah satu dari sedikit perusahaan-perusahaan Indonesia harus benar-benar terdaftar di bursa saham di luar Indonesia dan oleh karena itu, harus memenuhi standar internasional dan transparansi pemerintahan. Sementara Antam 65% diselenggarakan oleh pemerintah, Antam tidak dijalankan oleh birokrat. Antam tujuan utamanya adalah menciptakan nilai pemegang saham, tidak mengikuti arahan dari negara. Secara umum, pendekatan Antam untuk meningkatkan nilai pemegang saham adalah dengan menurunkan biaya operasi sementara menguntungkan berkembang secara berkelanjutan.

Salah satu cabang (*branch*) terletak di Bogor bagian barat yang berada di daerah Pongkor, Nanggung. Lokasi PT. Antam Tbk. Unit Bisnis Pertambangan Emas (UBPE) Pongkor secara administratif terletak di dalam wilayah Kecamatan

Nanggung, Leuwiliang dan Cigudeg, Kabupaten Bogor, Propinsi Jawa Barat. Sedangkan dari posisi geografi KP eksploitasi ini terletak pada koordinat 106□ 30' 01,0" - 106□ 35' 38,0" Bujur Timur dan 6º 36' 37,2" - 6º 48' 11,0" Lintang Selatan. Daerah KW 98 PP 0138 / Jabar tersusun oleh daerah pegunungan dengan ketinggian berkisar antara 300 meter sampai dengan 900 meter di atas permukaan air laut. Secara administratif, PT. Antam Tbk memiliki izin penambangan dalam bentuk lahan Kuasa Pertambangan (KP DU No. 562/Jabar). Di area ini telah ditemukan sepuluh unit Vein System yaitu Vein Cadas Copong, Cikoret, Cisuren, Gudang Handak, Pasir Jawa, Pasir Malang, Ciguha Utama, Kubang Cicau, Pamoyanan, dan Ciurug.

2.1.2. Visi dan Misi

Visi Antam 2020:

"Untuk menjadi perusahaan *global* yang berbasis pertambangan, dengan pertumbuhan yang sehat dan standar kelas dunia"

Misi Antam 2020:

* Untuk membangun dan menerapkan kelas dunia praktek terbaik dalam rangka untuk mengubah Antam menjadi pemain *global* yang benar-benar.

* Untuk menciptakan keunggulan operasional yang didasarkan pada biaya rendah operasi dan teknologi tepat guna dengan sangat perhatian terhadap kesehatan, keselamatan dan pelestarian lingkungan.

* Untuk memproses mineral yang ada cadangan dan deposito baru untuk meningkatkan keunggulan kompetitif perusahaan.

* Untuk mendorong pertumbuhan yang sehat melalui pengembangan bisnis yang berhubungan dengan pertambangan dan selektif diversifikasi dan integrasi untuk memaksimalkan pemegang saham *value*.

* Untuk meningkatkan karyawan kompetensi dan kesejahteraan, sementara mengembangkan budaya perusahaan yang mencakup kinerja luar biasa.

* Untuk berpartisipasi dalam upaya untuk meningkatkan kesejahteraan sosial masyarakat di sekitar daerah pertambangan khususnya di bidang pendidikan dan pemberdayaan ekonomi.

2.1.3. Badan Hukum

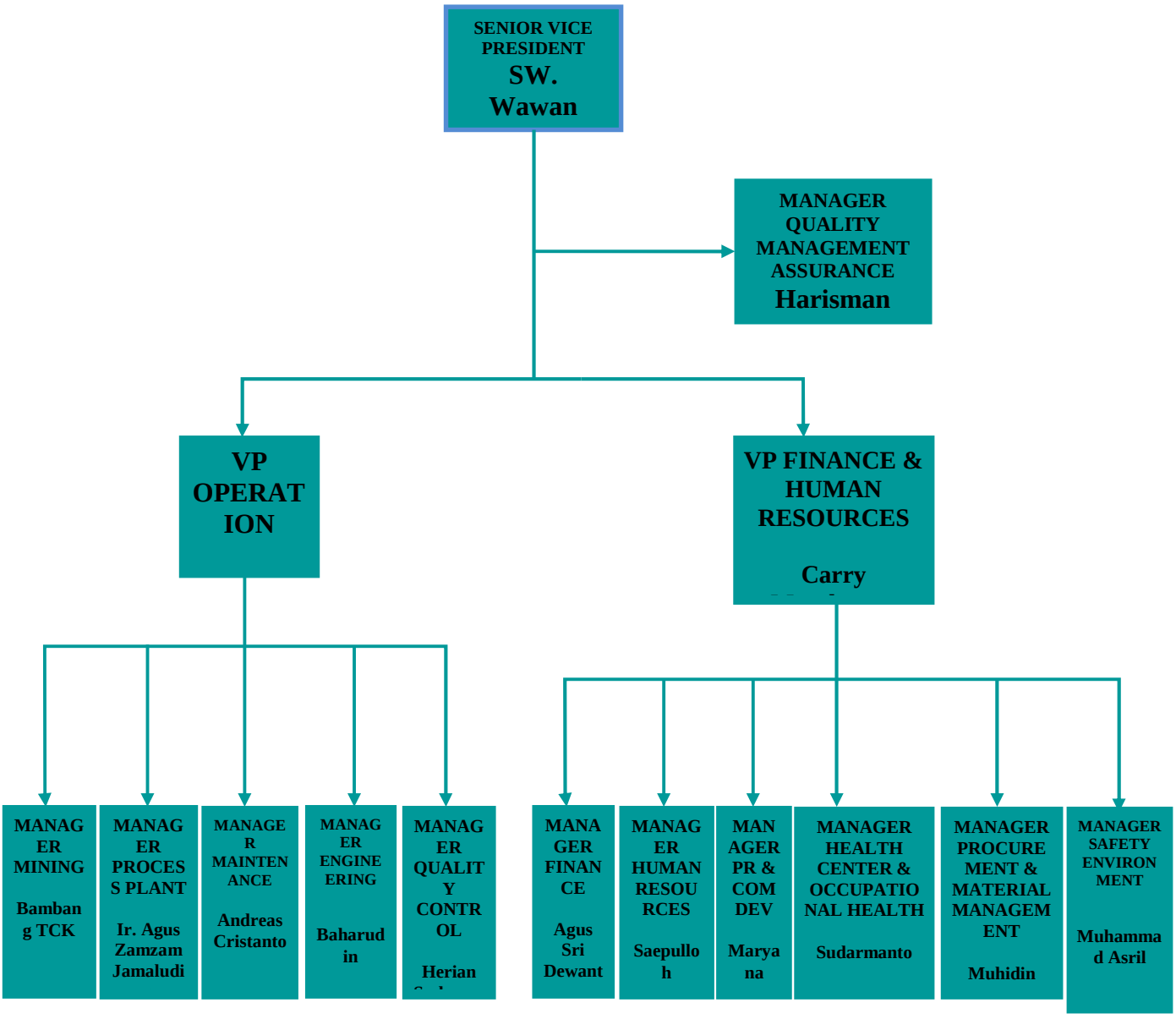
Menghasilkan produk-produk berkualitas tinggi yaitu nikel, emas dan mineral lain dengan selalu memperhatikan kelestarian lingkungan. Mencapai keunggulan kompetitif di pasar global bersandarkan pada kompetensi diri dengan tujuan untuk : memaksimalkan nilai pemegang saham; meningkatan kesejahteraan pegawai; dan meningkatan kesejahteraan masyarakat di sekitar daerah operasi pertambangan.

Komposisi Kepemilikan Saham Pemerintah :65.00%

Publik :35.00%

2.1.4. Struktur Organisasi

Dibawah ini merupakan struktur organisasi dari perusahaan Antam yang mencakup keseluruhan :



Gambar 2.1 Struktur Organisasi

2.2 Landasan Teori

2.2.1 Pengertian Sistem Informasi

Ada beragam pengertian mengenai sistem Informasi yang dikemukakan oleh beberapa para ahli [7]:

1. Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi [Alter (1992)].
2. Sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna [Bodnar dan Hopwood (1993)].
3. Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai. [Gelinas, Oram, dan Wiggins (1990)].
4. Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai. [Hall (2001)].
5. Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik. [Turban, McLean, dan Wetherbe (1999)].

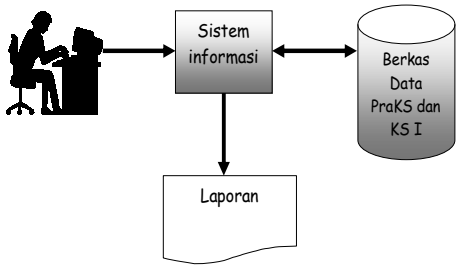
6. Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumberdaya (manusia, komputer) untuk mengubah masukan menjadi (informasi), guna mencapai sasaran perusahaan. [Wilkinson (1992)].

Dari berbagai definisi diatas, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, computer, tekhnologi informasi dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan.

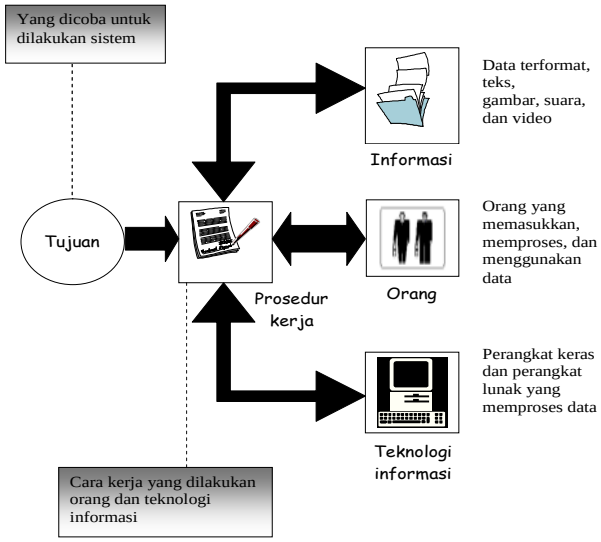
2.2.2 Konsep Dasar Sistem Informasi

Informasi merupakan hal yang sangat penting didalam mengambil keputusan. Informasi tersebut didapatkan dari sistem informasi (*information system*) atau disebut juga dengan *processing systems* atau *information processing systems* atau *information generating systems*. Sistem informasi adalah [6] :

"Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan".



Gambar 2.2 Sifat sistem informasi menggunakan komputer



Gambar 2.3 Siklus sistem informasi

2.2.3 Konsep Dasar Basis Data

Basis Data (*Database)* dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis Data menunjukkan suatu kumpulan data yang dipakai dalam suatu lingkungan perusahaan atau instansi-instansi. Penerapan basis data dalam sistem informasi disebut sistem basis data (*database system).*

1. Definisi Basis Data

Basis data (*database)* adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Secara sederhana database dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan computer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian pengaksesan dapat mencakup pemerolehan data maupun pemanipulasian data, sepeeti menambah dan menghapus data.

2. *Database Management System* (DBMS)

DBMS merupakan perangkat lunak atau program computer yang dirancang secara khusus untuk memudahkan pengelolaan database. Pengelolaan database secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak atau sistem yang khusus atau spesifik. Sistem ini yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Disamping itu sistem ini juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan atau konsistensi data dan sebagainya.

Perangkat lunak yang termasuk DBMS misalnya *dBase II+, dBase IV, FoxBase, RBase, MS-Access* dan *Borland-Paradox* atau *Borland-Interbase, MS-SQL Server, CA-Open Ingres, Oracle, Informix* dan *Sybase*.

3. Pengguna Basis Data

Ada beberapa jenis/tipe pemakai terhadap suatu sistem basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem :

a. Programmer Aplikasi (*Application Programmer*)

   Pemakai yang berinteraksi dengan basis data dengan menggunakan *Data Manipulation Language* (DML) untuk membuat aplikasi dengan menggunakan bahasa pemrograman.

b. *User* Khusus (*Specialized User*)

   Pemakai yang menulis aplikasi basis data non konvensional, tetapi untuk keperluan-keperluan khusus, seperti untuk aplikasi kecerdasan buatan., Sistem Pakar, Pengolahan Citra, dan lain-lain.

c. *User* Mahir (*Casual User*)

Pemakai yang berinteraksi dengan sistem tanpa membuat modul program.

d. *User* Umum (*Naïve User*)

Pemakai berinteraksi dengan aplikasi basis data yang telah dibuat atau disediakan oleh sistem.

4. Tujuan Basis Data

Tujuan awal dan utama dalam pengelolaan data dalam sebuah basis data adalah agar kita dapat memperoleh atau menemukan kembali data (yang kita cari) dengan mudah dan cepat. Tujuan basis data lain adalah kecepatan dan kemudahan (*Speed*), efisiensi Ruang Penyimpanan (*Space*), keakuratan (*Accuracy*), ketersediaan (*Availability*), kelengkapan (*Completeness*), keamanan (*Security*), kebersamaan pemakaian (*Sharability*).

2.2.4 Konsep Pemograman berorientasi objek

Ada beberapa tema yang mendasari tekhnologi berorientasi objek, mereka ini tidak unik pada sistem berorientasi objek tapi mereka sangat didukung pada metode analisis, perancangan serta implementasi dengan metodologi-metodologi berorientasi objek.

2.2.4.1 Kelas

Kelas membungkus (encapsulating) objek-objek. Suatu kelas tunggal dapat digunakan untuk menciptakan sejumlah objek-objek. Selain itu kelas juga masih bisa digunakan untuk menciptakan kelas-kelas lain yang mewarisi (*inheritance*) sebagian atau seluruh data serta fungsi yang dimiliki oleh kelas

yang disebutkan terdahulu. Selain itu kelas memiliki keadaan serta prilaku tertentu.

Pada awal perkembangannya, kelas didefinisikan sebagai tipe data abstrak (ADT – *Abstract Data Type)* [Simula67 dan Smalltalk]. Lalu tekhnologi berorientasi objek memperluas ide tentang ADT menjadi konsep kelas/objek dan menambah banyak informasi yang penting yang memungkinkan lebih mampu mengelola kompleksitas permasalahan

2.2.4.2 Abstraksi

Pada dasarnya abstraksi menemukan hal-hal esensial pada suatu objek dan mengabaikan hal-hal yang sifatnya insidential. Dengan kata lain menangkap sesuatu yang berarti untuk dituangkan dalam sistem/perangkat lunak alih-alih menangkap fakta yang ada. Pembangunan sistem informasi ini memfokuskan pada apa itu objek, apa cirri-ciri yang dimiliki objek(atribut-atributnya) serta apa yang objek lakukan (operasi-operasi apa yang dapat dilakukan oleh objek) sebelum memutuskan bagaimana ia dapat diimplementasikan.

2.2.4.3 Enkapsulasi

Pembungkusan (penyembunyian informasi) berarti meninggalkan aspek eksternal dari objek yang dapat dimasup(diakses) oleh objek lain dan memfokuskan diri pada implementasi internal suatu objek. Rincian implementasi internal dari suatu objek tersembunyi dari objek-objek lain dan terpisah dari implementasi *eksternal*, yaitu anatarmuka (*interface*) suatu objek dengan objek lain.

2.2.4.4 *Generalisasi* dan *polimorfisme*

Generalisasi/spesialisasi memungkinkan kelas-kelas berbagi data serta perilaku yang sama. Pada konteks pemograman, memungkinkan pengurangan ukuran kode dan menyediakan kemungkinan pengembangan sistem/perangkat lunak yang lebih mudah dipelihara. Generalisasi juga meningkatkan keterpaduan suatu kelas dan mengurangi derajat kesaling-bergantungan (meningkatkan koherensi) kelas saat implementasi kelak.

Polimorfisme mengijinkan penyesuaian berbagi kode untuk memenuhi keadaan tertentu. Dua mekanisme ini saling bekerja-sama untuk mendukung kemandirian objek individual serta mendukung proses pengembangan sistem/perangkat lunak berkelanjutan dimana komponen-komponen yang dibuat pada suatu aplikasi dapat digunakan pada aplikasi lain yang memerlukan layanan (*service)* yang sama.

2.2.4.5 Penggabungan Data (Atribut) dan Perilaku (Fungsi)

Pada sistem berorientasi objek, hierarki struktur data dan fungsi adalah identik pada hierarki pewarisan operasi. Artinya struktur data serta operasi-operasi pada setiap kelas akan diwariskan pada objek-objek dari kelas mana objek itu diciptakan serta diwariskan pula pada kelas-kelas secara hierarki berada dibawah kelas induk.

2.2.4.6 *Sharing*

Teknik pengembangan sistem berorientasi objek mengijinkan sharing (berbagi data/informasi serta berbagi penggunaan suatu operasi)pada beberapa

tingkatkan. Dalam konteks pemograman berorientasi objek, berbagi data dan prilaku memiliki beberapa tingkatan, seperti :

1. Metode serta peubah bersifat *Private*

   Metode serta data yang dikendalikan oleh suatu kelas tertentu dan tidak dapat diakses oleh objek-objek yang berbeda kelas.

2. Metode serta peubah bersifat *Public*

   Metode/operasi serta peubah/atribut yang bersifat *public* dapat diakses oleh semua objek-objek dalam aplikasi tertentu tanpa batasan apapun.

3. Metode serta peubah bersifat *Protected*

   Metode/operasi serta peubah/atribut yang bersifat *protected* hanya bisa diakses oleh kelas yang bersangkutan serta kelas-kelas yang merupakan turunannya mengikuti hierarki *generalisasi/spesialisasi*.

4. Metode serta peubah bersifat *Friend*

   Metode dan peubah dalam suatu kelas yang dapat diakses oleh kelas-kelas lain yang berhubungan dalam keadaan tertentu.

2.2.5 Tools dan analisis perancangan sistem

2.2.5.1 UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML mendefinisikan diagram-diagram berikut ini :

1. use case diagram

2. class diagram
3. behavior diagram :
   -- statechart diagram
   -- activity diagram
4. interaction diagram :
   -- sequence diagram
   -- collaboration diagram
5. component diagram
6. deployment diagram

2.2.5.1.1 Use case diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.

2.2.5.1.2 Class diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

2.2.5.1.3 Activity diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang

mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

2.2.5.1.4 Sequence diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

2.2.6 Jaringan Syaraf Tiruan

Artificial Intelligence adalah suatu pengetahuan yang membuat komputer dapat meniru kecerdasan manusia sehingga diharapkan komputer dapat melakukan hal-hal yang apabila dikerjakan manusia memerlukan kecerdasan.

JST atau Artificial Neural Network (ANN) adalah bagian dari sistem kecerdasan buatan (Artificial Intelligence, AI) yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah tiruan atau buatan dimaksudkan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. JST dimaksudkan untuk membuat model sistem komputasi yang dapat menirukan cara kerja jaringan syaraf biologis.

Model JST yang digunakan dalam penelitian untuk peramalan pendapatan Total tonase ini adalah Propagasi Balik (*Back Propagation*, BP). Baik tidaknya suatu model JST ditentukan oleh:

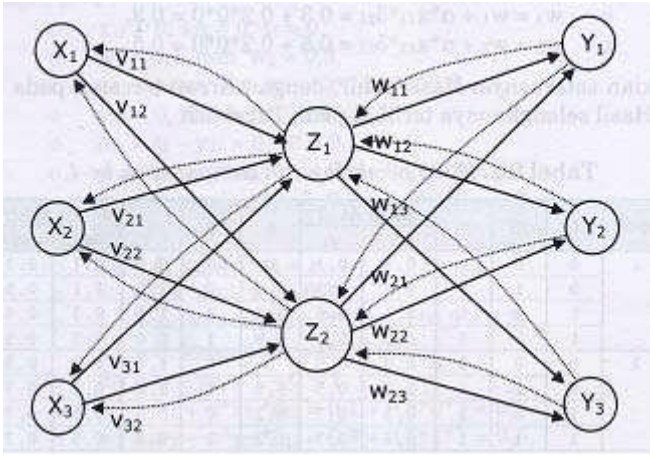
1. Pola antar neuron (arsitekur jaringan)
2. Metode untuk menentukan dan mengubah bobot (disebut metode *learning*)
3. Fungsi aktivasi

JST disebut juga: *brain metaphor*, *computational neuroscience*, *parallel distributed processing*. JST dapat belajar dari pengalaman, biasanya berhubungan dengan angka (numerik) sehingga data yang tidak numerik harus dibuat ke numeric, tidak ada rumus yang tetap (*fixed*) sehingga disebut dengan *free-estimator*, JST disebut *black box* atau tidak transparan karena tidak mampu menjelaskan bagaimana suatu hasil didapatkan. JST mampu menyelesaikan permasalahan yang tidak terstruktur dan sulit didefinisikan.

2.2.6.1 Backpropogation

Keunggulan yang utama dari sistem jaringan syaraf tiruan adalah kemampuan untuk “belajar” dari contoh yang diberikan. Algoritma pembelajaran yang digunakan adalah algoritma propagasi balik. Pada algoritma ini digunakan sinyal referensi dari luar (sebagai pengajar) dibandingkan dengan sinyal keluaran JST, hasilnya berupa sinyal kesalahan (error).

Dasar algoritma ini adalah memodifikasi bobot interkoneksi wjk pada jaringan sehingga sinyal kesalahan mendekati nol.



Gambar 2.4 Ilustrasi arsitektur JST propagasi balik

Algoritma pelatihan propagasi balik adalah sbb:

• Inisialisasi bobot.

• Kerjakan langkah-langkah berikut selama kondisi berhenti bernilai FALSE:

1. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

Feed-forward:

a. Tiap-tiap unit masukan (Xi, i=1,2,3,…,n) menerima sinyal masukan xi, dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).

b. Tiap-tiap unit tersembunyi (Zj, j=1,2,3,…,p) menjumlahkan sinyalsinyal masukan terbobot:

$z_in_j = v_{oj} + \sum_{i=1}^{n} Xi\ Vij$

Gunakan fungsi aktivasi untuk menghitung sinyal keluaran-nya:

$z_j = f(z_in_j)$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

c. Tiap-tiap unit keluaran (Yk, k=1,2,3,…,m) menjumlahkan sinyalsinyal masukan terbobot:

$$y_in_k = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}$$

gunakan fungsi aktivasi untuk menghitung sinyal keluaran-nya:

$y_k = f(y_in_k)$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

Propagasi Balik:

d. Tiap-tiap unit keluaran (Yk, k=1,2,3,…,m) menerima target referensi (berupa data target beban puncak), untuk dihitung informasi error-nya:

$\delta_k = (t_k - y_k)\, f'(y_in_k)$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}):

$\Delta w_{jk} = \alpha\, \delta_k\, z_j$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{ok}):

$\Delta w_{ok} = \alpha\, \delta_k$

kirimkan k δ ini ke unit-unit yang ada di lapisan bawahnya.

e. Tiap-tiap unit tersembunyi (Zj, j=1,2,3,…,p) menjumlahkan delta masukan-nya (dari unit-unit yang berada pada lapisan atasnya):

$\delta_in_j = \sum_{i=1}^{m} \delta_k \; w_{jk}$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error:

$\delta_j = \delta_in_j \; f'(z_in_j)$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai vij):

$\Delta v_{ij} = \alpha \, \delta_j \, x_i$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai voj):

$\Delta v_{oj} = \alpha \, \delta_j$

f. Tiap-tiap unit keluaran (Yk, k=1,2,3,…,m) memperbaiki bias dan bobotnya (j=1,2,3,…,p):

$w_{jk}\,(baru) = w_{jk}\,(lama) + \Delta w_{jk}$

Tiap-tiap unit tersembunyi (Zj, j=1,2,3,…,p) memperbaiki bias dan bobotnya (i=0,1,2,3,…,n):

v_{ij} (baru) = v_{ij} (lama) + Δv_{ij}

2. Tes kondisi berhenti.

2.2.7 Straight Line Method

Metode ini digunakan untuk mengalokasikan beban penyusutan tidak berdasarkan pada proporsi penggunaan aktiva yang sebenarnya. Dalam metode garis lurus lebih melihat aspek waktu daripada aspek kegunaan. Metode ini paling banyak diterapkan oleh perusahaan-perusahaan karena paling mudah diaplikasikan dalam akuntansi. Dalam metode penyusutan garis lurus, beban

penyusutan untuk tiap tahun nilainya sama besar dan tidak dipengaruhi dengan hasil/output yang diproduksi. Perhitungan tarif penyusutan untuk metode garis lurus adalah sebagi berikut:

Tarif Penyusutan = Harga Perolehan Nilai Sisa / Estimasi Umur Kegunaan

*Harga Perolehan Nilai Sisa= Nilai Beli-Nilai Residu

Metode penyusutan ini mempunyai kelebihan dan kelemahan.

Kelebihan dari metode ini adalah:

1) Mudah digunakan dalam praktek.
2) Lebih mudah dalam menentukan tarif penyusutan.

Kelemahan dari metode penyusutan ini adalah:

1) Beban pemeliharaan dan perbaikan dianggap sama setiap periode.
2) Manfaat ekonomis aktiva setiap tahun sama.
3) Beban penyusutan yang diakui tidak mencerminkan upaya yang digunakan dalam menghasilkan pendapatan.
4) Laba yang dihasilkan setiap tahun tidak menggambarkan tingkat pengembalian yang sesungguhnya dari umur kegunaan aktiva (dalam *matching principle*, beban penyusutan harus proporsional pada penghasilan yang dihasilkan).

2.2.8 Software Pendukung

2.2.8.1 Borland Delphi Versi 7.0

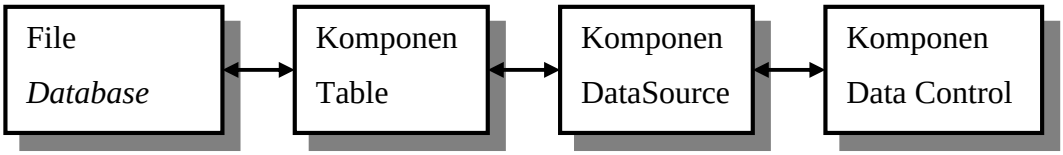
Delphi adalah suatu program berbasis bahasa Pascal yang berjalan dalam lingkungan *Windows*. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut RAD yang telah membuat pemrograman menjadi lebih mudah. Delphi

adalah suatu bahasa pemrograman yang telah memanfaatkan metode pemrograman *Object Oriented Programming* (OOP).

Borland Delphi atau yang biasa disebut Delphi saja, merupakan sarana aplikasi visual. Delphi merupakan generasi penerus dari Turbo Pascal, Turbo Pascal yang diluncurkan pada tahun 1983 dirancang untuk dijalankan pada sistem operasi DOS (yang merupakan sistem operasi yang paling banyak digunakan pada saat itu). Sedangkan Delphi yang diluncurkan pertama kali tahun 1995 dirancang untuk beroperasi dibawah sistem operasi *Windows*.

Kelebihan Borland Delphi 7.0 yaitu telah dilengkapi dengan sejumlah komponen yang tergolong sebagai *dbExpress*, yang memungkinkan koneksi ke MySQL ataupun Oracle dilakukan dengan mudah, sehingga Delphi dapat digunakan sebagai aplikasi *front-end* yang berhubungan dengan *database server*.

Database dengan Delphi menggunakan konsep seperti gambar dibawah ini:



Gambar 2.5 Bagan Konsep *Database*

Keterangan :

1. File ***Database*** : File *database* dari sistem *database* lain seperti Dbase (*.dbf), Paradox (*.db), Microsoft Access (*.mdb), mySql (*.sql) dan lain-lain.

2. Komponen ***Table*** : Komponen yang mewakili file *database*. Setiap melakukan proses dalam komponen tabel tersebut, maka isi file *database* yang terkoneksi ke komponen tersebut berubah juga.
3. Komponen ***DataSource*** : Komponen penghubung antara komponen tabel dengan komponen data control. Dalam *datasource* harus diisi tabel yang berelasi ke *datasource* tersebut.
4. Komponen-Komponen **Data *Control*** : Komponen yang digunakan untuk menampilkan data-data yang berasal dari *datasource* (tabel). Data *control* ada yang berbentuk tabel, label, *editbox*, gambar, *combobox, listbox* dan lain-lain.

2.2.8.2 MySQL

SQL adalah sebuah konsep pengoprasian *database*, terutama untuk pemilihan/seleksi dan pemasukan data, yang memungkinkan pengoprasian data dikerjakan dengan mudah secara otomatis. Salah satu aplikasi *database* yang menggunakan bahasa SQL adalah MySQL. MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public Lisence*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source*/komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama SQL dalam *database* sejak lama. MySQL memiliki beberapa kelebihan diantaranya :

1. MySQL sebagai *Data Management System* (DBMS) dan *Relation Databese Management System* (RDBMS).

2. MySQL adalah *software database* yang *OpenSource*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
3. MySQL merupakan *database server*, jadi dengan menggunakan *database* ini dapat menghubungkan ke media *internet* sehingga dapat diakses jarak jauh.
4. MySQL merupakan sebuah *database* client. Selain menjadi *server* yang melayani permintaan, MySQL juga dapat melakukan *query* yang mengakses *database* pada *server*, jadi MySQL dapat juga berperan sebagai *Client*.
5. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.
6. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *Gigabyte* sekalipun.
7. MySQL didukung oleh driver ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
8. MySQL adalah *database* menggunakan enkripsi *password*. Jadi *database* ini cukup aman karena memiliki *password* untuk mengaksesnya.
9. MySQL dapat menciptakan lebih dari 16 kunci per tabel dan dalam satu kunci memungkinkan berisi belasan *field*. Serta MySQL mendukung *field* yang dijadikan sebagai kunci *primer* dan kunci unik atau (*Unique*).
10. MySQL memiliki kecepatan dalam pembuatan tabel maupun peng-*update*-an tabel.

SQL (*Structure Query Language*) dibagi menjadi 2 bentuk *Query*, yaitu :

1. DDL (*Data Definition Language*)

DDL adalah sebuah metode *query* SQL yang berguna untuk mendefinisikan data pada sebuah *database*, adapun *query* yang dimiliki adalah :

a. *CREATE* : untuk melakukan pembuatan tabel dan *database*

b. *DROP* : untuk melakukan penghapusan tabel maupun *database*

c. *ALTER* : untuk melakukan pengubahan struktur tabel yang telah dibuat, baik menambah *field* (*add*), mengganti mana *field* (*change*) ataupun menamakannya kembali (*rename*), serta menghapus (*drop*).

2. DML (*Data Manipulation Language*)

DML adalah sebuah metode *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan pemanipulasian *database* yang telah ada atau telah dibuat sebelumnya. Adapun *query* yang termasuk didalamnya adalah :

a. *INSERT* : untuk melakukan penginputan/pemasukan data pada tabel *database*.

b. *UPDATE* : untuk melakukan perubahan/peremajaan terhadap data yang ada pada tabel.

c. *DELETE* : untuk melakukan penghapusan data pada tabel. Penghapusan ini dapat dilakukan secara sekaligus (seluruh isi tabel) maupun hanya beberapa *recordset*.

Perintah-Perintah SQL

1. Menambah Data (*Insert*)

   Perintah SQL yang digunakan untuk menambah data ke *database* adalah perintah *INSERT*. Perintah ini mempunyai bentuk umum sebagai berikut:

   INSERT [INTO] nama_tabel [(daftar *field*/kolom)] VALUES (daftar_data)

2. Memilih/Mengakses Data (*Select*)

   Perintah SQL yang digunakan untuk memilih data adalah perintah *SELECT*. Bentuk umum perintah ini adalah sebagai berikut:

   SELECT daftar_select FROM daftar_tabel

   [WHERE kondisi_pencarian]

   [GROUP BY daftar_group_by]

   [HAVING kondisi_pencarian]

   [ORDER BY daftar_order[ASC │ DESC]]

3. Mengedit/Mengubah Data (*Update*)

   Perintah SQL yang digunakan untuk mengubah data adalah perintah *UPDATE SET*. Perintah ini mempunyai bentuk umum sebagai berikut:

   UPDATE nama_table SET *field*1=databaru1 [,data2=databaru2]

   [WHERE kondisi_update]

4. Menghapus Data (*Delete*)

   Perintah SQL yang digunakan untuk menghapus data adalah perintah *DELETE FROM*. Perintah ini mempunyai bentuk umum sebagai berikut:

   DELETE FROM tabel_hapus WHERE kondisi_hapus