BAB II

LANDASAN TEORI

2.1 Sejarah Singkat PT. Industri Telekomunikasi Indonesia (Persero)

Dari cikal bakal Laboratorium Penelitian & Pengembangan Industri Bidang Pos dan Telekomunikasi (LPPI-POSTEL), pada 30 Desember 1974 berdirilah PT. Industri Telekomunikasi Indonesia (INTI) sebagai Badan Usaha Milik Negara (BUMN) dengan misi untuk menjadi basis dan tulang punggung pembangunan Sistim Telekomunikasi Nasional (SISTELNAS).

Sejak berdirinya sampai sekarang ini, PT. Industri Telekomunikasi Indonesia (Persero) berkedudukan di Kompleks Laboratorium PTT di Jl. Moch Toha No. 77 serta di Jl. Moch Toha 225 Bandung. Seiring waktu dan berbagai dinamika yang harus diadaptasi, seperti perkembangan teknologi, regulasi, dan pasar, maka selama lebih dari 30 tahun berkiprah dalam bidang telekomunikasi, INTI telah mengalami berbagai perubahan dan perkembangan.

Pada Era 1974 – 1984, fasilitas produksi yang dimiliki INTI antara lain adalah:

1. Pabrik Perakitan Telepon
2. Pabrik Perakitan Transmisi
3. Laboratorium Software Komunikasi Data
4. Pabrik Konstruksi & Mekanik

Kerjasama Teknologi yang pernah dilakukan pada era ini antara lain dengan Siemen, BTM, PRX, JRC, dan NEC. Pada era tersebut produk Pesawat Telepon

Umum Koin (PTUK) INTI menjadi standar Perumtel (sekarang Telkom). Fasilitas produksi terbaru yang dimiliki INTI pada era 1984 - 1994, di samping fasilitas-fasilitas yang sudah ada sebelumnya, antara lain adalah Pabrik Sentral Telepon Digital Indonesia (STDI) pertama di Indonesia dengan teknologi produksi Trough Hole Technology (THT) dan Surface Mounting Technology (SMT).

Kerjasama Teknologi yang pernah dilakukan pada era ini antara lain adalah:

1. Bidang sentral (switching), dengan Siemens
2. Bidang transmisi dengan Siemens, NEC, dan JRC
3. Bidang CPE dengan Siemens, BTM, Tamura, Shapura, dan TatungTEL

Pada era ini, PT. Industri Telekomunikasi Indonesia memiliki reputasi dan prestasi yang signifikan, yaitu:

1. Menjadi pionir dalam proses digitalisasi sistem dan jaringan telekomunikasi di Indonesia.
2. Bersama Telkom telah berhasil dalam proyek otomatisasi telepon di hampir seluruh ibu kota kabupaten dan ibu kota kecamatan di seluruh wilayah Indonesia.

Pada era 1994 – 2000 selama 20 tahun sejak berdiri, kegiatan utama INTI adalah murni manufaktur. Namun dengan adanya perubahan dan perkembangan kebutuhan teknologi, regulasi dan pasar, INTI mulai melakukan transisi ke bidang jasa *engginering*.

Pada masa ini aktivitas manufaktur di bidang switching, transmisi, CPE dan mekanik-plastik masih dilakukan. Namun situasi pasar yang berubah, kompetisi yang makin ketat dan regulasi telekomunikasi yang makin terbuka menjadikan

posisi INTI di pasar bergeser sehingga tidak lagi sebagai market leader. Kondisi ini mengharuskan INTI memiliki kemampuan *sales force* dan *networking* yang lebih baik. Kerjasama teknologi masih berlangsung dengan Siemens secara single-source.

Pada era 2000 - 2004 kerjasama teknologi tidak lagi bersifat single source, tetapi dilakukan secara multi source dengan beberapa perusahaan multinasional dari Eropa dan Asia. Aktivitas manufaktur tidak lagi ditangani sendiri oleh INTI, tetapi secara spin-off dengan mendirikan anak-anak perusahaan dan usaha patungan, seperti:

1. Bidang CPE, dibentuk anak perusahaan bernama PT. INTI PISMA International yang bekerja sama dengan JITech International, bertempat di Cileungsi Bogor.
2. Bidang mekanik dan plastik, dibentuk usaha patungan dengan PT. PINDAD bernama PT. IPMS, berkedudukan di Bandung.
3. Bidang-bidang switching, akses dan transmisi, dirintis kerja sama dengan beberapa perusahaan multinasional yang memiliki kapabilitas memadai dan adaptif terhadap kebutuhan pasar. Beberapa perusahan multinasional yang telah melakukan kerjasama pada era ini, antara lain:
    a. SAGEM, di bidang transmisi dan selular
    b. MOTOROLA, di bidang CDMA
    c. ALCATEL, di bidang *fixed & optical access network*
    d. Ericsson, di bidang akses
    e. Hua Wei, di bidang switching & akses

Dari serangkaian tahapan restrukturisasi yang telah dilakukan, INTI pada era 2005 – sekarang, kini memantapkan langkah transformasi mendasar dari kompetensi berbasis manufaktur ke *engginering solution*. Hal ini akan membentuk INTI menjadi semakin adaptif terhadap kemajuan teknologi dan karakteristik serta perilaku pasar.

Dari pengalaman panjang INTI sebagai pendukung utama penyediaan infrastruktur telekomunikasi nasional dan dengan kompetensi sumber daya manusia yang terus diarahkan sesuai proses transformasi tersebut, saat ini INTI bertekad untuk menjadi mitra terpercaya di bidang penyediaan jasa profesional dan solusi total yang fokus pada Infocom System & Technology Integration (ISTI).

2.2 Manajemen Proyek

Manajemen adalah proses merencanakan, mengorganisasikan, memimpin, mengendalikan usaha-usaha anggota organisasi dan proses penggunaan sumber daya organisasi untuk mencapai tujuan-tujuan organisasi yang telah ditetapkan. Proyek adalah suatu rangkaian pekerjaan yang diadakan dalam selang waktu tertentu dan mempunyai tujuan khusus. Yang membedakan proyek dengan pekerjaan lain adalah sifatnya yang khusus dan tidak bersifat rutin pengadaannya, sehingga pengelolaannyapun memerlukan perhatian ekstra lebih banyak. Sedangkan Manajemen Proyek adalah proses pengelolaan proyek yang meliputi perencanaan, pengorganisasian dan pengaturan tugas – tugas serta sumber daya

yang dimiliki untuk mewujudkan tujuan yang ingin dicapai, dengan mempertimbangakan faktor – faktornya terutama waktu dan biaya.

Masalah penjadawalan proyek merupakan suatu proses pengalokasian waktu pengerjaan proyek dan tenaga kerja. Untuk menentukan proyek yang harus dikerjakan dengan jangka waktu yang disediakan, sehingga terbentuk susunan antara tenaga kerja, waktu penyelesaian proyek, dan jumlah proyek yana baik, dalam hal ini agar tidak terjadi bentrok dalam pengaturan jadwal tenaga kerja proyek satu dengan proyek lain dan menghindari keterlambatan dalam proses penyelesaian proyek.

Tenaga kerja dan proyek yang dikerjakan akan dijadikan sebagai sebuah *event* dan waktu penyelesaian proyek yang disediakan untuk mejadwalankan *event* – *event* tadi disebut dengan *timeslot.* Inti dari penjadwalan proyek ini adalah bagaimana memilih pasangan *event* dan *timeslot (e,t)* yang selanjutnya akan dimasukan ke dalam jadwal.

2.3 Definisi Penjadwalan

Secara umum penjadwalan dapat didefinisikan sebagai proses pengalokasian sumber-sumber dalam jangka waktu tertentu untuk melakukan sekumpulan pekerjaan. Definisi ini mengandung dua arti yang berbeda, yaitu :

1. Penjadwalan merupakan fungsi pengambilan keputusan, yaitu menentukan jadwal.
2. Penjadwalan merupakan suatu teori, yaitu sekumpulan prinsip-prinsip dasar, model-model, teknik-teknik, dan kesimpulan-kesimpulan logis

dalam proses pengambilan keputusan yang memberikan dalam fungsi penjadwalan (nilai konseptual).

Penjadwalan adalah proses pengurutan pembuatan produk secara menyeluruh pada beberapa mesin. Menurut Morton dan Pentico penjadwalan adalah proses pengorganisasian, pemilihan dan pemberian waktu dalam penggunaan sumber dayanya untuk melaksanakan aktivitas yang diperlukan dalam menghasilkan output yang diinginkan dengan memenuhi waktu yang diinginkan pula.

Persoalan penjadwalan timbul apabila jumlah mesin dan peralatan yang dimiliki terbatas sedangkan terdapat beberapa pekerjaan yang dapat dikerjakan secara bersama. Untuk mendapat hasil yang optimal dengan keterbatasan sumber daya yang dimiliki, maka diperlukan adanya penjadwalan sumber-sumber tersebut secara efisien.

Tujuan penjadwalan secara umum adalah :

1. Meningkatkan produktivitas mesin, yaitu dengan mengurangi waktu menganggur mesin.
2. Mengurangi terhadap persediaan barang setengah jadi, dengan mengurangi rata-rata pekerjaan yang menunggu dalam antrian karena mesin sibuk oleh pekerjaan lain.
3. Mengurangi keterlambatan (*tardiness*). Dalam banyak hal, beberapa atau semua pekerjaan mempunyai batas waktu penyelesaian (*duedate*). Apabila suatu pekerjaan melewati batas waktu tersebut, maka akan dikenai pinalti.

4. Keterlambatan dapat diperkecil dengan mengurangi maksimal *tardiness* atau mengurangi pekerjaan yang terlambat (*number of tardy job*).

Terdapat target utama yang ingin dicapai melalui penjadwalan *flow shop* ini yaitu jumlah output yang dihasilkan (*throughput*) berupa *makespan*. Penjadwalan *flow shop* didefinisikan sebagai penjadwalan dimana setiap *job* mempunyai pola aliran atau rute proses yang tetap pada seluruh mesin.

2.3.1 Klasifikasi Masalah Penjadwalan *Flow Shop*

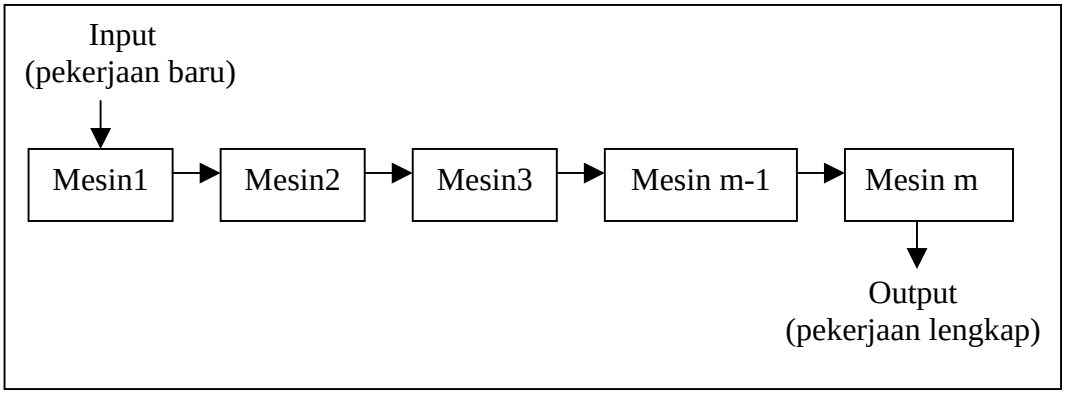
Model penjadwalan dapat dibedakan menjadi 4 jenis keadaan, yaitu :

1. Mesin yang digunakan, dapat berupa proses dengan mesin tunggal atau proses dengan mesin majemuk.
2. Pola aliran proses, dapat berupa aliran identik atau sembarang.
3. Pola kedatangan pekerjaan, Statis atau Dinamis.
4. Sifat informasi yang diterima, dapat berupa Deterministik atau Stokastik.

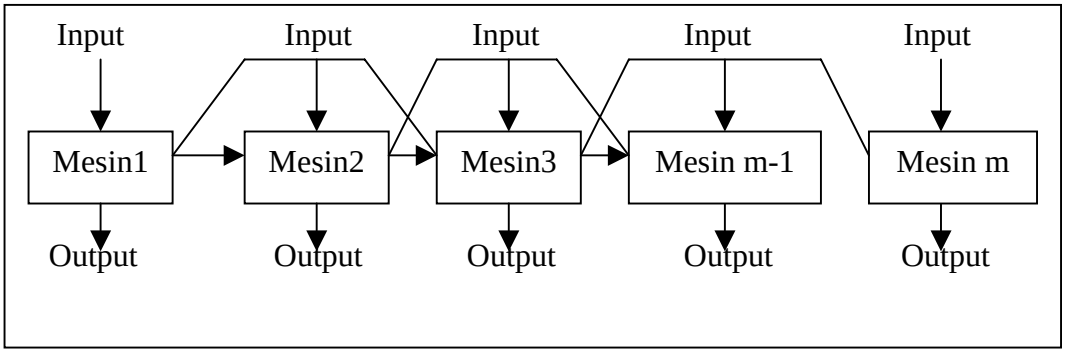
Pada jenis keadaan pertama, jumlah mesin dapat dibedakan atas mesin tunggal dan mesin majemuk. Model mesin tunggal merupakan model dasar dan biasanya dapat diterapkan dalam kasus mesin majemuk.

Pada model kedua, pola aliran dapat dibedakan atas *Flow Shop* dan *Job Shop*. Pada *Flow Shop* dijumpai pola aliran pemrosesan dari suatu mesin ke mesin yang lain dalam urutan (*routing*) tertentu. Semua pekerjaan yang mengalir pada saat produksi yang sama tanpa boleh melewatinya disebut dengan *pure Flow Shop*. Tetapi jika pekerjaan yang datang kedalam *Flow Shop* tidak harus dikerjakan pada semua mesin, jenis *Flow Shop* ini disebut dengan *General Flow*

Shop. Contoh pola aliran *Pure Flow Shop* dan contoh pola aliran *Gereral Flow Shop* ditunjukkan pada Gambar 2.1 dan Gambar 2.2.

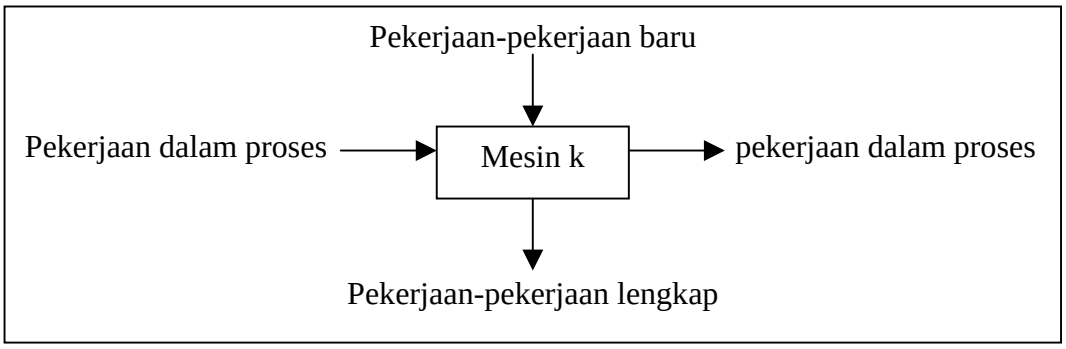


Gambar 2.1 *Pola Aliran Pure Flow Shop*



Gambar 2.2 *Pola Aliran General Flow shop*

Pada *Job Shop* setiap pekerjaan mempunyai *routing* yang berbeda. Alir proses yang tidak searah ini mengakibatkan setiap pekerjaan yang akan diproses pada suatu mesin dapat merupakan pekerjaan baru atau pekerjaan yang sedang dikerjakan (*work in proses*).



Gambar 2.3 *Pola Aliran Job Shop*

Pada model ketiga, pola kedatangan pekerjaan dapat dibedakan atas pola kedatangan Statis dan Dinamis. Pada pola Statis, pekerjaan datang secara bersamaan pada waktu nol, siap dikerjakan pada mesin-mesin yang juga sudah siap untuk bekerja atau kedatangan pekerjaan yang tidak bersamaan tetapi saat kedatangan telah diketahui sejak waktu nol. Sedangkan pola Dinamis mempunyai kedatangan pekerjaan tidak menentu, dijumpai adanya variable waktu sebagai faktor pengaruh.

Pada model keempat, perilaku elemen-elemen penjadwalan dapat dibedakan atas Deterministik dan Stokastik. Model Deterministik dapat dilihat dari adanya kepastian atas informasi tentang beberapa aspek. Sedangkan pada model Stokastik, mengandung unsur ketidakpastian. Aspek yang dimaksud adalah :

1. Karakteristik pekerjaan dari segi kedatangan, jumlah (kuantitas) pekerjaan, batas waktu penyelesaian (*duedate*) dan perbedaan kepentingan antar pekerjaan.
2. Karakteristik pekerjaan dari segi banyaknya operasi, susunan mesin dan waktu proses.

3. Karakteristik mesin dari segi jumlah dan kapasitas mesin, kemampuan dan kecocokan tiap mesin dengan pekerjaan yang diberikan.

2.3.2 Beberapa Istilah dalam Penjadwalan *Flow Shop*

Penjadwalan *Flow shop* dapat dijelaskan sebagai berikut. Jika terdapat n *job* {*j*1, *j*2, …, *j*n}, maka harus diproses pada m mesin {m1, m2, …, mm}. Waktu yang diperlukan untuk memproses *job* ***i*** pada mesin ***j*** adalah t_{ij}. Jadi permasalahan penjadwalan adalah menentukan urutan *job* yang memberikan solusi terbaik berdasarkan kriteria tertentu.

Beberapa istilah yang digunakan dalam masalah penjadwalan yaitu :

1. Waktu proses (*processing time* = $\mathbf{t_j}$) : yaitu rentang waktu yang dibutuhkan untuk menyelesaikan suatu operasi pada *job* **j**.
2. *Ready Time* ($\mathbf{r_j}$)**,** yaitu saat mulai suatu *job* **j** dapat dikerjakan.
3. *DueDate* ($\mathbf{d_j}$), yaitu batas waktu akhir suatu *job* harus sudah terselesaikan. Bila melewati batas ini, suatu *job* dikatakan terlambat (*tardy*)
4. Waktu penyelesaian (*completion time* = $\mathbf{C_j}$) : saat *job* **j** telah selesai dikerjakan.
5. Waktu tinggal (*flow time* = $\mathbf{F_j}$) : lamanya *job* **j** berada dilantai pabrik (*shop*). Flow time dihitung sejak *job* siap dijadwalkan sampai *job* selesai dikerjakan.
6. *Lateness* ($\mathbf{L_j}$), yaitu merupakan penyimpangan waktu penyelesaian saat *job* terhadap *duedate job* yang bersangkutan. *Lateness* dihitung dengan persamaan $L_j = C_j - d_j$.

   $L_j < 0$, saat penyelesaian memenuhi batas akhir (*earliness*).

$L_j > 0$, saat penyelesaian melewati batas akhir (*tardiness*).

7 *Slack* (**SL_j**), yaitu waktu yang tersedia bagi suatu pekerjaan.

$SL_j = d_j - t_j$.

8 *Tardiness* (**T_j**), yaitu merupakan keterlambatan penyelesaian suatu *job* terhadap *duedate job* tersebut.

$T_j = \max \{0, L_j\}$.

9 *Makespan* (**Ms**), yaitu waktu dimana semua pekerjaan terakhir selesai ($MaxL_j$).

2.3.3 Aturan Prioritas Penjadwalan (*Priority Dispatching Rules*)

Penentuan prioritas (*Priority Dispatching*) dari pekerjaan (*job*) dalam pendekatan penjadwalan *heuristic* adalah untuk memilih satu dari beberapa operasi yang mengalami gangguan. Penentuan prioritas tersebut sangat ditentukan dari tujuan yang ingin dicapai.

Beberapa aturan prioritas diberikan menurut French (1980), adalah :

1. **Random**

   Pemilihan operasi secara random artinya setiap pekerjaan kemungkinan yang sama untuk dipilih.

2. **Most Work Remaining (MWR)**

   Prioritas tertinggi diberikan pada operasi dengan sisa waktu proses terlama. Aturan prioritas ini cocok digunakan untuk menghasilkan jadwal dengan *makespan* terpendek.

3. **First-Come-First-Serve (FCFS)**

   Prioritas diberikan kepada pesanan yang tiba lebih dahulu disumber. Aturan ini jika konsumen mementingkan waktu pelayanan.

4. **Earliest Due Date (EDD)**

   Prioritas diberikan kepada pesanan dengan *due date* terpendek. Aturan ini dapat berjalan dengan baik jika batas waktu proses *job* relatif sama.

5. **Shortest Processing Time (SPT)**

   Prioritas diberikan kepada pesanan dengan waktu proses yang lebih kecil.

6. **Fewest Operation (FO)**

   Prioritas tertinggi diberikan pada produk dengan sisa operasi paling sedikit. Aturan ini biasanya mengurangi WIP, waktu penyelesaian rata-rata produk (*mean flow time*) dan rata-rata keterlambatan (*mean lateness*).

7. **Shortest Total Processing Time Remaining (STPT)**

   Prioritas tertinggi diberikan kepada pesanan yang mempunyai waktu sisa yang lebih kecil.

8. **Slack Time (ST)**

   Prioritas tertinggi diberikan pada produk dengan slack time lebih kecil. ST = dd (*duedate*)- Pd (*Present date*) – TOTR (*Total Operation Remaining*).

9. **Critical Ratio (CR)**

   Prioritas diberikan kepada pesanan dengan CR yang lebih kecil. CR adalah perbandingan antara waktu yang tersisa sebelum saat kirim dibagi dengan waktu yang tersisa.

2.3.4 Penjadwalan *Flow Shop*

Pada teknik penjadwalan *Flow Shop* dapat dibagi ke dalam 2 teknik, yaitu :

1. **Teknik Optimasi**
   a. Algoritma Johnson untuk *makespan*.
   b. Algoritma Branch and Bound.
2. **Teknik Heuristik**

Karakteristik *Flow Shop* yang utama adalah adanya aliran mesin yang searah, waktu pengerjaan *job* dapat dilihat dengan membuat *routing*.

2.4 Algoritma Genetika

Algoritma Genetika adalah algoritma heuristik yang didasarkan atas mekanisme evolusi biologis. Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah yang terbentuk adaptasi (alami atau buatan) dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom.

Algoritma genetika menggunakan mekanisme seleksi alam dan genetika alam. Hal ini berarti istilah – istilah yang terdapat pada algoritma genetika harus sesuai dengan istilah pada seleksi alam dan genetika alam. Dalam genetika alam, *komosom* terdiri dari susunan *gen – gen.* Tiap gen mengandung nilai atau sifat tertentu yang disebut *allele,* sedang posisi *gen* dalam *kromosom* disebut *lokus.* Selanjutnya, satu atau beberapa *kromosom* bergabung membentuk paket genetika yang disebut *fenotif.* Dalam genetika buatan *kromosom bersesuaian* dengan *string*

yang dibentuk dari beberapa karakter, setiap karakter ini mempunyai posisi *(lokus)* dan mengandung nilai tertentu *(allele).* Satu atau beberapa *string* akan bergabung membentuk struktur *(geneotif),* bila struktur tersebut dikodekan akan diperoleh satu titik yang merupakan salah satu altenatif solusi *(fenotif).* Untuk jelasnya, terminologi yang digunakan dalam genetik alami dan padanannya dalam algoritma genetika dapat dilihat pada tabel 2.1

Tabel 2.1 Triminologi dalam genetik alam dan algoritma genetika

Genetik Alam	**Algoritma Genetika**
Kromosom	String
Gen	Karakter
Allele	Nilai karakter
Locus	Posisi dalam string
Geneotif	Struktur
Fenotif	Paramater

Algoritma genetika merupakan algoritma pencarian yang bekerja berdasarkan mekanisme seleksi alam untuk menetukan individu – individu yang berkualitas tinggi yang terdapat dalam sebuah domain yang disebut populasi. Pencarian dilakukan dengan melaksanakan sebuah prosedur interaktif untuk mengatur sebuah populasi individu yang merupakan kandidat – kandidat solusi. Dalam siklus iterasi terdapat dua tahap yaitu proses seleksi dan rekombinasi.

Tahap seleksi dilakukan dengan mengavaluasi setiap kualitas setiap individu dalam populasi untuk mendapatkan hasil evaluasi. Selanjutnya dipilih individu-individu yang mengalami *rekombinasi*. Proses pemilihan individu-individu ini dilakukan secara acak dan individu-individu yang mempunyai kualitas lebih baik mempunyai kemungkinan yang lebih besar untuk dipilih.

Tahap *rekombinasi* meliputi proses genetik untuk mendapatkan populasi baru dari calon-calon individu yang diperoleh pada tahap seleksi. Anggota populasi baru diperoleh dengan menerapkan operator genetik (reproduksi, pindah silang dan mutasi) secara acak pada calon-calon individu yang terpilih dalam tahap seleksi. Pada tahap *rekombinasi* ini akan diperoleh individu-individu baru yang berbeda dengan individu induknya dan dengan demikian akan diperoleh domain pencarian baru.

2.4.1 Komponen Utama Algoritma Genetika

Ada 6 komponen dalam Algoritma Genetika, yaitu :

2.4.1.1 Representasi Kromosom

Gen merupakan bagian dari kromosom. Satu gen biasanya mewakili satu variabel. Gen dapat direpresentasikan ke dalam bentuk : string, bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program atau representasikan lainnya yang dapat diimplementasikan untuk operator genetika.

2.4.1.2 Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetik yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak,

namun demikian harus tetap memperhatikan dominasi dan solusi kendala permasalahan yang ada.

2.4.1.3 Fungsi Evaluasi

Ada dua hal yang harus dilakukan dalam melakukan evaluasi kromosom yaitu : Fungsi obyektif (fungsi tujuan) dan konversi fungsi obyektif ke dalam fungsi *fitness*. Secara umum fungsi *fitness* diturunkan dari fungsi obyektif dengan nilai yang tidak negatif, maka perlu ditambahkan suatu konstanta *x* agar nilai *fitness* yang terbentuk menjadi tidak negatif.

2.4.1.4 Seleksi

Ada beberapa metode seleksi yang digunakan dalam Algoritma Genetika, yaitu :

1. *Roulette Wheel Selection*

Metode ini menirukan permainan *Roulette Wheel* dimana masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*nya. Kromosom yang memiliki nilai *fitness* yang lebih besar menempati potongan lingkaran yang lebih besar dibandingkan dengan kromosom bernilai *fitness* rendah.

2. *Rank Selection*

Pada *Rank Selection*, yang pertama dilakukan adalah meranking populasi sehingga setiap kromosom akan menerima nilai *fitness* 1, kedua terburuk bernilai

2 dan seterusnya, dan kromosom terbaik akan memiliki nilai *fitness* N (sejumlah kromosom dalam populasi).

3. *Steady-state Selection*

Cara kerja *Steady-state Selection* adalah sebagai berikut. Pada setiap generasi beberapa kromosom dengan nilai *fitness* tertinggi dipilih untul membentuk *offspring* baru. Kemudian beberapa kromosom yang *fitness*nya terendah dibuang dan *offspring* baru menggantikan tempat mereka. Populasi yang *survive* akan menuju ke generasi selanjutnya.

4. *Elitism*

Elitism adalah suatu metode yang berguna untuk mengcopy kromosom terbaik (atau beberapa kromosom terbaik) pada suatu populasi dan mempertahankannya di populasi yang baru. Sisa populasi yang lain dibangun dengan cara diatas.

2.4.1.5 Operator Genetik

Operator Genetik merupakan cara kerja yang dilakukan dalam proses Algoritma Genetika untuk dapat menghasilkan individu baru yang lebih baik dari sebelumnya.

2.4.1.5.1 *Crossover*

Pada proses ini akan dipilih secara acak dua individu dan tempat pertukaran, dimana akan menandai kromosom yang berbeda antara kedua tempat pertukaran akan bertukar tempat satu sama lain.

2.4.1.5.2 Mutasi

Mutasi adalah operator yang menghasilkan perubahan acak secara spontan pada kromosom. Pada dasarnya akan mengubah secara acak nilai suatu bit pada posisi tertentu, dengan mengganti bit 1 dengan bit 0, atau sebaliknya. Pada mutasi sangat dimungkinkan muncul kromosom baru yang semula belum muncul dalam populasi awal. Faktor terbesar dalam teori evolusi yang menyebabkan suatu kromosom dapat bertahan hidup, mati, melakukan persilangan atau melakukan mutasi adalah lingkungan.

2.4.2 Parameter Algoritma Genetika

Yang dimaksud dengan parameter adalah parameter kontrol Algoritma Genetika. Nilai parameter ini ditentukan juga berdasarkan permasalahan yang akan dipecahkan.

2.4.2.1 Probabilitas *Crossover* (Pc)

Menunjukan rasio dari anak yang dihasilkan dalam setiap generasi dengan ukuran populasi. Jika *crossover* tidak ada, maka *offspring* akan meniru dari induknya. Tetapi jika *crossover* ada, *offspring* dibuat dari bagian kromosom induk (jika kemungkinan *crossover* adalah 100% maka semua *offspring* dibuat oleh *crossover*, tetapi jika 0% maka *offspring* dibuat dengan meniru kromosom dari populasi lama secara keseluruhan).

2.4.2.2 Probabilitas Mutasi (Pm)

Menunjukan presentasi jumlah total gen pada populasi yang akan mengalami mutasi. Jika tidak ada mutasi, *offspring* langsung mengambil dari bentuk *crossover* (meniru) dengan tidak merubah lagi. Jika mutasi dilakukan, bagian kromosom diubah (jika kemungkinan 100%, seluruh kromosom akan diubah dan jika 0% tidak ada yang diubah). Mutasi dibuat untuk mencegah penurunan kualitas dari kromosom yang telah didapat.

2.4.2.3 Population Size

Merupakan kumpulan kromosom dari populasi (1 generasi). Dari setiap generasi ukuran populasi harus sama.

2.4.3 Cara Kerja Algoritma Genetika Sederhana

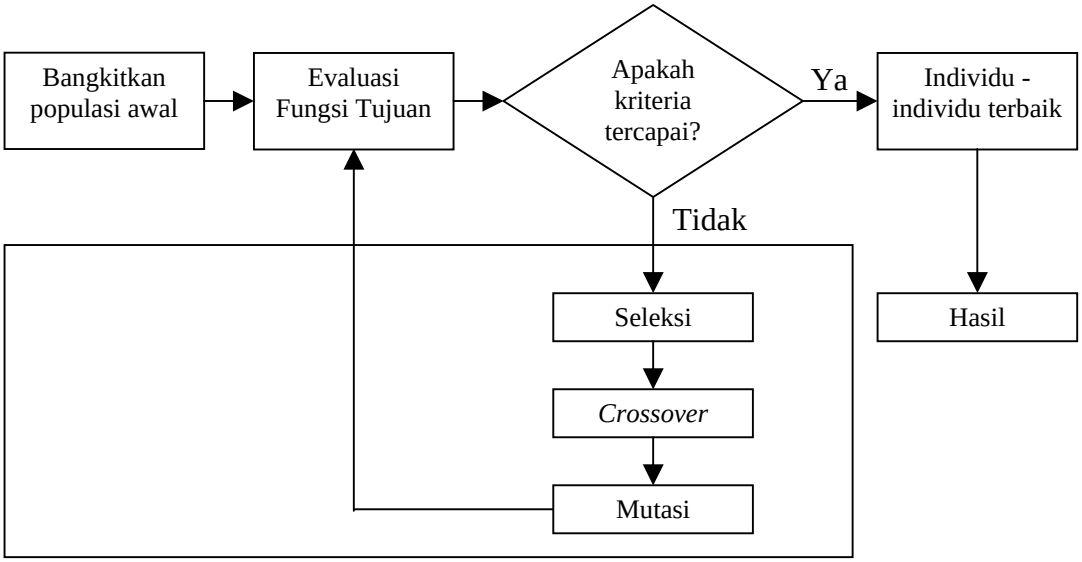
Secara sederhana Algoritma Genetika terdiri dari langkah-langkah :

1. Generasi = 0 (generasi awal)
2. Inisialisasi *populasi* awal, P(generasi) secara acak
3. Evaluasi nilai *fitness* pada setiap individu dalam P(generasi)
4. Kerjakan langkah-langkah berikut hingga generasi mencapai Maksimal generasi.
    a. Generasi = generasi + 1 (tambah generasi)
    b. Seleksi *populasi* tersebut untuk mendapatkan kandidat induk, P’(generasi)
    c. Lakukan *crossover* pada P’(generasi)

d. Lakukan Mutasi pada P’(generasi)

e. Lakukan evaluasi *fitness* setiap individu pada P’(generasi)

f. Bentuk populasi baru: P(generasi) = {P(generasi-1) yang *survive*, P’(generasi)}



Gambar 2.4 Diagram Alir Algoritma Genetika Sederhana

2.5. Aplikasi dan Form pada Borland Delphi 7

Aplikasi merupakan penerapan, pengimplementasian suatu permasalahan pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk menerapkan permasalahan tersebut, sehingga menjadi suatu bentuk program tanpa menghilangkan nilai-nilai dasar dari permasalahan yang ada. Aplikasi dapat dibedakan menjadi dua yaitu Aplikasi *Windows* dan Aplikasi *Konsol.* Aplikasi *Windows* adalah aplikasi yang berjalan pada *Windows.* Aplikasi non-*Windows* contohnya yang berjalan pada DOS yang biasa disebut aplikasi *Konsol.*

Secara umum, sebuah aplikasi paling sedikit melibatkan sebuah *form.* Namun biasabya sebuah aplikasi melibatkan banyak *form.* Sebuah *form* umumnya melibatkan komponen lain kotak *Combo Box* dan Tombol *Radio Button* merupakan contoh komponen tersebut. Tetapi tidak semua komponen terlihat secara visual, komponen yang terlihat sevara visual biasa disebut dengan kontrol pada Delphi. Sebuah aplikasi akan diletakkan pada sebuah *Project*. Sebuah *project* dapat membawahi beberapa *form. Form* merupakan inti dari sebuah aplikasi pada Delphi karena dianggap sebagai dasar aplikasi *Windows.*

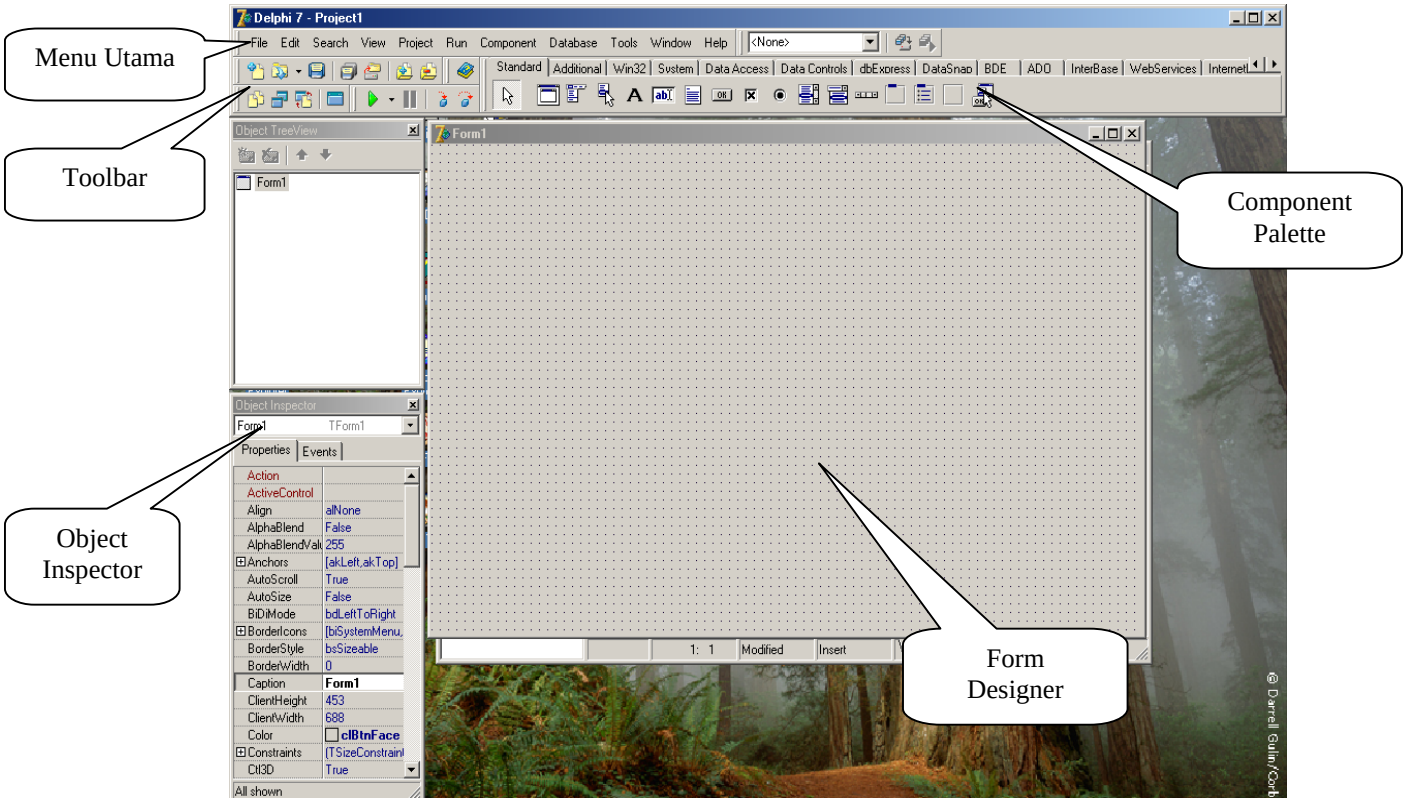
2.5.1 Gambaran Singkat Bagian-bagian *IDE*

IDE *(Intergrated Development Environment)* adalah sebuah lingkungan dimana semua *tools* yang diperlukan untuk mendesain, menjalanklan dan mengetes sebuah aplikasi yang terjadi dan terhubung baik sehingga memudahkan program. Melalui *IDE* ini pemprograman sevara visual merancang tampilan untuk pemakai (antar muka pemakai) dan menuliskan kode.

2.5.2 Kelebihan-kelebihan Delphi

Membangun perangkat lunak ini dilakukan dalam lingkungan pengembangan terpadu *(Intergrated Development Environment)* Delphi 7 dari Borland. Pemilihan *tools* pengembangan ini didasarkan pada beberapa pertimbangan sebagai berikut :

1. Delphi mendukung pemrograman berorientasi objek. Dengan demikian, hasil analisis dan desain berorientasi objek yang telah dilakukan pada tahap sebelumnya dapat dimplementasinkan dengan mudah.
2. Delphi memiliki kelas-kelas yang telah terdefinisi, hal ini membuat pemrograman tidak perlu dilakukan dari awal. Pemrograman cukup dilakukan dengan menggunakan kembali atau menurunkan kelas-kelas baru dari kelas yang sudah ada.
3. Delphi dalam pembangunan kelas-kelas kontrol antarmuka visual yang telah terdefinisi. Hal ini memudahkan dalam pembangunan perangkat lunak dengan antarmuka grafis yang *User Friendly*.
4. Delphi memiliki fasilitas *Borland Database Engine (BDE)* yang memberikan lingkungan pemrograman basis data tang tranparan dengan kecepatan pengaksesan yang jauh lebih baik.



Gambar 2.5 Komponen utama dalam Borland Delphi 7