BAB II
TINJAUAN PUSTAKA

2.1 Profil Telstar Bandung

2.1.1 Sejarah singkat Telstar

Sekitar sembilan tahun yang lalu tepatnya tanggal 14 Agustus 2001 sebuah lembaga pendidikan Bahasa Inggris TELSTAR Lahir, Djoko Setiadi adalah pendiri lembaga tersebut. Setelah mengelola Pendidikan Bahasa Inggris terbesar di Indonesia yakni Oxford Course Indonesia lebih dari 28 Tahun. Djoko Setiadi ingin mengembangkan sebuah lembaga pendidikan dengan program yang lebih spesifik dengan metode pengajaran Bahasa Inggris yang disesuaikan dengan kebutuhan peserta.

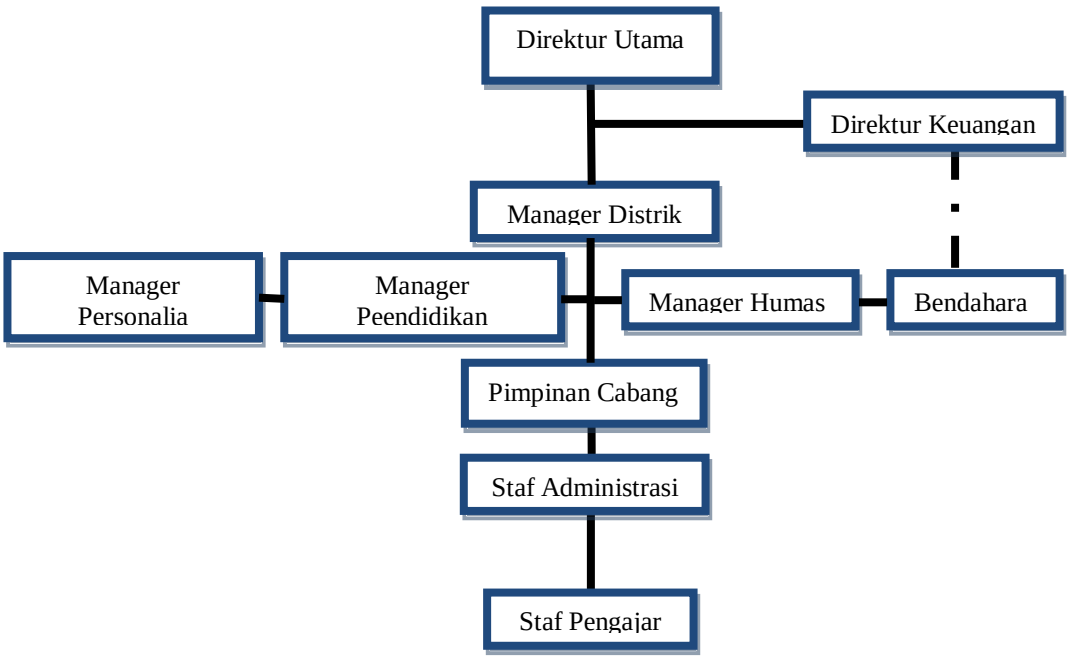
Dengan didukung oleh beberapa rekan-rekan yang berpengalaman tentang pengajaran Bahasa Inggris dan Ahli di bidang pengembangan kurikulum dari dalam negeri dan luar negeri, maka tekat untuk mewujudkan lembaga tersebut lahirlah nama TELSTAR. Singkatan dari *The English Language Skill Training And Refreshing* yang bermakna bahwa meningkatkan kemampuan berbahasa Inggris dengan baik dan benar melalui Pelatihan / *Training* atau *Refreshing*. Untuk pertama kalinya Telstar berdiri Bulan Agustus 2001 Di Daerah Pinggiran kota, tepatnya di Jl. Soreang - Banjaran 364 Bandung. Kini TELSTAR mempunyai tiga cabang dengan kantor pusat di Jl. Natasari VI H. 71 Telp. 5400553, 76448449, 76441155. Email : telstar_course@yahoo.com.

2.1.2 Logo Telstar dan Struktur Organisasi

Serperti halnya lembaga lain, Telstar pun memiliki logo yang ditunjukan pada gambar 2.1 dan pada gambar 2.2 struktur organisasi di lembaga pendidikan bahasa inggris Telstar :



Gambar 2.1 Logo Telstar



Gambar 2.2 Struktur Organisasi

2.1.3 Job description

Adapun tugas dari setiap bagian di lembaga pendidikan Telstar adalah sebagai berikut :

1. Direktur Utama

   Tugasnya adalah mengkoordinasikan serta mengawasi semua pekerjaan yang dilaksanakan oleh pafra manager yang dilaksanakan oleh Manager Distrik (Kepala Wilayah). Koordinasi dan pengawasan ini bertujuan untuk memonitor sejauh mana perkembangan dan kemajuan yang telah di capai dari program program yang telah dicanangkan sebelumnya. Selain itu Direktur Utama juga memimpin rapat tahunan (*annual meeting*) para manager distrik dan pimpinan cabang.

2. Direktur Keuangan

   Bertugas mengkoordinasikan serta mengawasi arus masuk dan arus keluar uang perusahaan dan juga memonitor pendapatan dan pengeluaran setiap cabang berdasarkan laporan keuangan yang dikirimkan oleh para bendahara setiap bulan. Dalam pelaksanaan kerjanya Direktur keuangan dibantu oleh beberapa tenaga keuangan yang bekerja dengan system komputisasi.

3. Manager Personalia

   Manager Personalia memiliki tugas dan weweang dalam penanganan masalah yang berkaitan dengan sumberdaya manusia (*human resources*) seperti kebutuhan terhadap tenaga pengajar / tenaga manajemen, peraturan

pegawai, standar gaji, penerimaan pegawai, serta pengadaan program latihan (*training programme*) bagi para pegawai baru.

4. Manajer Humas

   Peran yang dimainkan oleh seorang kepala hubungan masyarakat dalam suatu perusahaan amat vital karena ia tidak hanya melakukan hubungan ke luar tetapi juga hubungan ke dalam perusahaan. Sama halnya dengan peran yang dimainkan oleh manajer hubungan masyarakat Telstar.

5. Manajer Pendidikan

   Berwenang dalam penyusunan materi-materi pendidikan untuk membuat buku-buku panduan bagi para siswa. Selain itu juga berwenang dalam penyempurnaan kerikulum pendidikan dalam penyusunan materi-materi pendidikan tersebut. Beberapa staf khususnya para tenaga pengajar tetap Telstar membantu manajer pendidikan ini. Biasanya materi pendidikan yang telah disusun dievaluasi atau ditinjau kembali setiap setahun sekali agar seluruh materi tersebut tetap sesuai degan perkembangan zaman dan kebutuhan para peserta pendidikan Telstar.

6. Manajer Distrik

   Bertugas memimpin, mengatur, serta mengawasi setiap kegiatan lembaga-lembaga yang berada di bawah yayasan pendidikan Telstar yang berada ditingkat provinsi. Bila di dalam suatu provinsi Tersebut hanya terdapat satu buah cabang maka disebut sebagai pemimpin cabang bukan kepala wilayah. Manajaer distrik atau kepala cabang dinentuk oleh seorang manajer pendidikan wilayah atau cabang serta beberapa tenaga pengajar

tetap dan seorang bendahara. Manajer distrik berwenang untuk menerima pegawai baru apabila memang diperlukan baik tenaga kependidikan maupun sebagai administrasi.

7. Bendahara

   Tugasnya adalah wajib memberikan laporan keuangan harian kepada direktur keuangan untuk dibuat menjadi laporan keuangan mingguan dan bulanan.

8. Pimpinan cabang

   Bertugas dalam membantu para tutor dalam melaksanakan kurikulum yang telah ditetapkan oleh Telstar pusat serta melakukan pemeriksaaan hasil ujian para peserta kursus yang diselenggarakan oleh Telstar.

2.2 Landasan Teori

2.2.1 Pengertian Algoritma Genetik

Algoritma genetika merupakan algoritma heuristik atau pendekatan yang menerapkan proses seleksi alam dan evolusi yang dikemukakan oleh Charles Darwin. Algoritma genetik pertama kali diperkenalkan oleh John Holland pada tahun 1960-an dan pengembangan lebih lanjut dilakukan oleh Holland bersama murid-murid dan rekan sekerjanya di Universitas Michigan pada tahun 1960-an dan 1970-an.

Sebagai ide dasar heuristik, algoritma genetika dimulai dengan sekumpulan solusi yang direpresentasikan dengan kromosom, disebut populasi. Semua solusi dari satu populasi diambil dan digunakan untuk membentuk populasi baru. Hal ini dimotivasi dengan harapan populasi baru akan lebih baik daripada populasi lama.

Solusi terpilih yang merupakan orangtua pembentuk keturunan (offspring), dipilih berdasarkan lingkungannya atau nilai fitness, semakin baik maka semakin besar kesempatan yang dimiliki untuk bertahan dan bereproduksi. Hal ini dilakukan berulang kali hingga memenuhi beberapa kondisi penghentian, misalnya jumlah generasi atau tercapainya hasil yang dicari.

2.2.2 Prosedur Algoritma Genetik

Sifat dari algoritma genetika adalah mencari kemungkinan-kemungkinan dari kandidat solusi untuk mendapatkan suatu solusi yang optimal bagi penyelesaian masalah. Ruang cakupan dari semua solusi yang layak (feasible) yaitu obyek-obyek diantara solusi yang sesuai dinamakan ruang pencarian (search space). Tiap titik dalam ruang pencarian merepresentasikan satu solusi yang layak yang dapat ditandai dengan nilai fitness yang sesuai dengan masalah. Solusi yang dicari dalam algoritma genetika adalah titik (1 atau lebih) di antara solusi yang layak dalam ruang pencarian. Sifat dari pencarian inilah yang menyebabkan algoritma genetika baik diterapkan untuk menyelesaikan masalah NP-complete.

Istilah fungsi evaluasi dan fungsi fitness sering saling dipertukarkan penggunaannya. Akan tetapi, adalah berguna untuk membedakan kedua fungsi tersebut dalam algoritma genetika. Fungsi evaluasi, atau fungsi obyektif, merupakan suatu pengukuran performansi terhadap sekumpulan parameter tertentu. Fungsi fitness mentransformasikan pengukuran performansi tersebut dalam suatu alokasi kesempatan bereproduksi. Evaluasi terhadap suatu barisan yang menyatakan sekumpulan parameter tertentu adalah independen dari evaluasi

terhadap barisan-barisan lain. Fitness suatu barisan selalu didefinisikan berdasarkan anggota-anggota lain dalam populasi. Secara umum fitness didefinisikan *fi/f* dengan *fi* adalah fungsi evaluasi dengan kromsom i dan f adalah evaluasi rata-rata dari semua kromosom dalam populasi.

Secara sederhana, algoritma umum dari algoritma genetik ini dapat dirumuskan menjadi beberapa langkah, yaitu:

1. [**Mulai**] Membangun populasi random sebanyak n kromosom (sesuai dengan masalahnya)
2. [**Fitness**] Evaluasi setiap fitness f(x) dari setiap kromosom x pada populasi
3. [**Populasi baru**] Membuat populasi baru dengan mengulang langkah-langkah berikut sampai populasi lengkap.
    1. [**Seleksi**] Pilih 2 kromosom orangtua dari populasi berdasarkan fitnessnya (semakin besar fitnessnya semakin besar kemungkinannya untuk terpilih)
    2. [**Perkawinan silang**] Sesuai dengan besarnya kemungkinan perkawinan silang, orangtua terpilih disilangkan untuk membentuk anak. Jika tidak ada perkawinan silang, maka anak merupakan salinan dari orangtuanya.
    3. [**Mutasi**] Sesuai dengan besarnya kemungkinan mutasi, anak dimutasi pada setiap lokus (posisi pada kromosom).
    4. [**Penerimaan**] Tempatkan anak baru pada populasi baru.
4. [**Ganti**] Gunakan populasi yang baru dibentuk untuk proses algoritma selanjutnya.

5. [**Tes**] Jika kondisi akhir terpenuhi, berhenti, dan hasilnya adalah solusi terbaik dari populasi saat itu.
6. [**Ulangi**] Ke nomer 2.

2.2.2.1 Pengkodean

Pengkodean adalah cara untuk merepresentasikan masalah kedalam bentuk kromosom. Ada beberapa cara pengkodean dan pemilihannya tergantung dari masalah yang akan diselesaikan. Berikut ini adalah beberapa jenis pengkodean yang sering digunakan.

2.2.2.1.1 Pengkodean biner (*binary encoding*)

Pengkodean biner adalah pengkodean yang paling umum digunakan karena algoritma genetika awalnya menggunakan pengkodean ini. Pada pengkodean biner setiap kromosom merupakan barisan bit 0 atau 1, seperti dapat dilihat pada gambar berikut ini :

Tabel 2.1 contoh kromosom dengan pengkodean biner

Kromosom A	10110010110010101110010 1
Kromosom B	11111110000011000001111 1

Pengkodean biner memberikan banyak kemungkinan kromosom meskipun hanya memiliki jumlah allele yang sedikit, 1 atau 0. Tetapi pengkodean ini sering tidak sesuai untuk beberapa masalah dan kadang harus dilakukan pengkoreksian setelah perkawinan silang dan mutasi.

Contoh masalah yang sesuai menggunakan pengkodean biner adalah masalah ransel (knapsack), dimana terdapat beberapa jenis benda dengan ukuran dan berat yang berbeda sedangkan kapasitas ransel terbatas. Permasalahannya adalah memilih sejumlah barang untuk memaksimumkan nilai dari barang-barang tersebut di dalam ransel, tetapi total ukuran barang-barang tersebut tidak melebihi dari ukuran ransel itu sendiri. Masalah ransel ini dapat dienkodekan dengan barisan bit, dimana jumlah bit menyatakan jumlah barang yang ada, sehingga setiap bit berhubungan dengan setiap benda yang ada di dalam ransel. Bit 1 berarti benda ada di dalam ransel sedangkan 0 berarti benda tidak berada di dalam ransel.

2.2.2.1.2 Pengkodean permutasi (*permutation encoding*)

Pengkodean permutasi digunakan dalam masalah yang memerlukan pengurutan data (ordering problem), seperti masalah wiraniaga (travelling salesman problem), atau masalah pengurutan tugas (task ordering problem). Pada pengkodean ini setiap kromosom merupakan barisan angka yang merepresentasikan angka pada urutan.

Tabel 2.2 Contoh kromosom dengan pengkodean permutasi

Kromosom A	1 5 3 2 6 4 7 9 8
Kromosom B	8 5 6 7 2 3 1 4 9

Pengkodean permutasi hanya berguna pada masalah pengurutan. Perkawinan silang dan mutasi yang dilakukan pada pengkodean ini harus menghasilkan kromosom yang memiliki konsistensi urutan.

Contoh masalah yang dapat diselesaikan dengan pengkodean ini adalah masalah wiraniaga. Terdapat sejumlah kota dan jarak diantara semua kota itu. Seorang wiraniaga harus mengunjungi semua kota itu, tetapi sebisa mungkin dia tidak perlu melakukan terlalu banyak perjalanan, sehingga harus ditemukan urutan kota yang akan dikunjungi dengan jumlah total jarak yang minimal. Masalah ini dapat dienkodekan dengan pengkodean permutasi dimana setiap angka dalam kromosom mewakili setiap kota yang harus dikunjungi dalam urutan.

2.2.2.1.3 Pengkodean nilai (*value encoding*)

Pengkodean nilai dapat digunakan pada masalah yang mempunyai nilai yang rumit, dimana nilai yang dienkodekan langsung merupakan representasi dari masalah. Contohnya adalah pengkodean data yang menggunakan angka riil yang akan menjadi susah jika menggunakan pengkodean biner karena setiap angka riil tidak dapat direpresentasikan secara tepat hanya dalam bentuk bit 1 atau 0.

Tabel 2.3 Contoh kromosom dengan pengkodean nilai

Kromosom A	1.2324, 5.3243, 0.4556, 2.3293, 2.4545
Kromosom B	A, B, D, J, E, I, F, J, D, H, D, I, E, R, J
Kromosom C	(back), (back), (right), (forward), (left)

Dalam pengkodean nilai, setiap kromosom merupakan barisan dari beberapa nilai. Nilai bisa berupa apa saja, seperti bilangan biasa, bilangan riil, karakter sampai dengan obyek-obyek yang rumit.

Pengkodean nilai sangat baik untuk beberapa masalah khusus, dilain sisi perkawinan silang dan mutasi yang diperlukan untuk pengkodean ini seringkali memerlukan pengembangan tersendiri yang lebih spesifik pada masalah.

Contoh masalah yang menggunakan pengkodean ini adalah masalah pencarian bobot pada jaringan saraf (neural network). Terdapat sebuah jaringan saraf dengan arsitekturnya. Permasalahnya adalah mencari bobot masukan dari neuron-neuron untuk melatih jaringan supaya memberikan keluaran yang diinginkan. Masalah ini dapat dienkodekan menggunakan pengkodean nilai bilangan riil, dengan gen-gen dalam genome menyatakan korespondensi bobot-bobot pada masukan untuk jaringan saraf.

2.2.2.1.4 Pengkodean pohon (*tree encoding*)

Pengkodean pohon lebih banyak digunakan untuk menyusun program atau ekspresi, bagi pemrograman genetika (genetic programming).

Dalam pengkodean pohon, setiap kromosom merupakan pohon dari sejumlah obyek, seperti fungsi atau perintah di dalam bahasa pemrograman. Pengkodean pohon sangat baik bagi penyusunan program. Bahasa pemrograman LISP sering digunakan untuk melakukan pengkodean pohon, karena program di dalamnya dapat direpresentasikan ke dalam bentuk ini dan dapat dengan mudah dilakukan parse menjadi sebuah pohon, sehingga perkawinan silang dan mutasi dapat dilakukan relatif lebih mudah.

Contoh masalah yang direpresentasikan dengan pengkodean pohon adalah masalah mencari fungsi berdasarkan nilai-nilai yang diberikan. Diberikan beberapa nilai masukan dan keluaran. Permasalahannya adalah mencari fungsi yang akan memberikan nilai keluaran terbaik (dekat dengan yang diinginkan)

untuk semua masukan. Masalah ini dapat dienkodekan ke dalam pengkodean pohon dengan kromosom merupakan fungsi-fungsi yang dinyatakan dalam pohon.

2.2.3 Operator dalam Algoritma Genetika

Operator-operator pada algoritma genetika yang sering digunakan adalah seleksi, perkawinan silang dan mutasi. Semua operator tersebut memegang peranan penting bagi keberhasilan algoritma genetika dalam menyelesaikan masalah.

Pemilihan jenis operator yang akan digunakan tergantung dari masalah yang akan diselesaikan. Hal yang harus diperhatikan adalah menghindari terjadinya konvergensi prematur, yaitu mencapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah solusi optimum lokal. Selain itu, untuk menjamin bahwa seluruh ruang pencarian dapat dijangkau oleh alogritma genetika, perbedaan (diversity) diantara kromosom dalam populasi merupakan hal yang penting.

2.2.3.1 Seleksi

Seleksi merupakan proses pemilihan individu/ kromosom dari generasi lama untuk dijadikan orangtua yang akan saling dikawin silang untuk membentuk individu baru di generasi baru. Yang menjadi permasalahan adalah bagaimana cara memilih kromosom untuk dijadikan orangtua. Berdasarkan teori evolusi Darwin, kromosom yang terbaik seharusnya dapat bertahan hidup dan membentuk

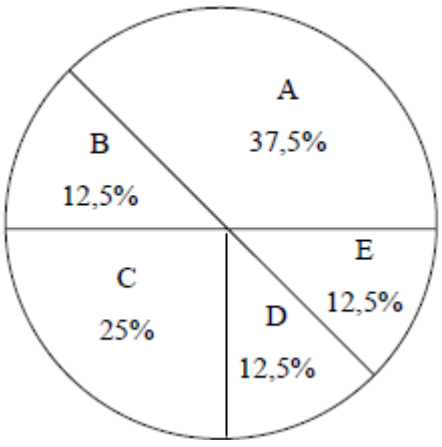
keturunan baru. Ada beberapa jenis metode seleksi, berikut ini adalah beberapa metode seleksi yang sering digunakan.

2.2.3.1.1 Seleksi roda roulette (*roulette wheel selection*)

Pada metode seleksi ini, orangtua dipilih berdasarkan nilai fitnessnya, semakin baik nilai fitnessnya maka semakin besar kemungkinannya untuk terpilih. Diandaikan semua kromosom diletakkan pada sebuah roda roulette, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai fitnessnya seperti pada contoh berikut :

Tabel 2.4 Contoh populasi dengan 5 kromosom beserta fitnessnya

Kromosom	Fitness
A	15
B	5
C	10
D	5
E	5



Gambar 2.3 Probabilitas terpilihnya suatu kromosom dalam roda *roulette*

Probabilitas masing-masing individu merupakan hasil pembagian antara fitness masing-masing individu dengan total fitness dalam populasi. Dari contoh di atas,

kromosom A memiliki probabilitas 37,5% untuk terpilih sebagai orangtua pembentuk keturunan baru pada setiap pemilihan kromosom (setiap roda diputar).

Skema seleksi dengan roda roulette ini adalah berdasarkan skala fitness (fitness scale). Karena terpilihnya suatu kromosom dalam populasi untuk dapat berkembang biak adalah sebanding dengan fitnessnya, maka akan terjadi semacam tradeoff antara ekplorasi dan ekplotasi jika terdapat satu atau kelompok kecil kromosom yang mempunyai fitness yang baik yaitu antara mengeksplorasi bagian-bagian baru dalam ruang pencarian, atau terus mengeksplotasi informasi yang telah diperoleh. Kecenderungan kromosom yang baik untuk terpelihara terus dapat membawa ke hasil optimum lokal atau konvergensi dini (premature convergence) ke suatu hasil yang bukan optimum global. Sebaliknya jika semua kromosom dalam populasi mempunyai fitness yang hampir sama, maka seleksi ini akan menjadi seleksi yang bersifat acak.

2.2.3.1.2 Seleksi rangking (*rank selection*)

Metode seleksi roda roulette akan memiliki masalah ketika terdapat perbedaan fitness yang jauh. Sebagai contoh, jika fitness kromosom terbaik adalah 90 % dari semua roda roulette dapat menyebabkan kromosom yang lain memiliki kesempatan yang sangat kecil untuk dapat terpilih.

Seleksi ranking terlebih dahulu merangking atau mengurutkan kromosom di dalam populasi berdasarkan fitnessnya kemudian memberi nilai fitness baru berdasarkan urutannya. Kromosom dengan fitness terburuk akan memiliki fitness baru bernilai 1, terburuk kedua bernilai 2 dan seterusnya, sehingga kromosom

yang memiliki fitness terbaik akan memiliki nilai fitness N, dimana N adalah jumlah kromosom di dalam populasi.

Sebagai ilustrasi dapat dilihat pada gambar berikut yang juga merupakan contoh dari metode seleksi roda roulette tadi :

Tabel 2.5 Keadaan sebelum dirangking

Kromosom	Fitness
A	15
B	5
C	10
D	5
E	5

Tabel 2.6 Keadaan setelah dirangking

Kromosom	Fitness	Fitness Baru
B	5	1
D	5	2
E	5	3
C	10	4
A	15	5

Setelah proses pengurutan dan pemberian nilai fitness baru, setiap kromosom akan memiliki kesempatan yang lebih adil untuk terpilih. Tetapi metode ini dapat menyebabkan konvergensi menjadi lambat, karena kromosom terbaik tidak terlalu berbeda dengan yang lainnya.

2.2.3.1.3 Seleksi turnamen (*tournament selection*)

Seleksi turnamen merupakan variasi antara seleksi roda roulette dan seleksi rangking. Sejumlah k kromosom tertentu dari populasi beranggota n kromosom ($k \leq n$) dipilih secara acak dengan probabilitas yang sama. Dari k kromosom yang terpilih kemudian akan dipilih satu kromosom dengan fitness

terbaik, yang diperoleh dari hasil pengurutan rangking fitness semua kromosom terpilih. Perbedaannya dengan seleksi roda roulette adalah pemilihan kromosom yang akan digunakan untuk berkembang biak tidak berdasarkan skala fitness dari populasi.

Untuk k = 1, seleksi turnamen akan menjadi sama dengan seleksi secara acak karena hanya melibatkan satu kromosom. Untuk k = 2, maka akan dipilih dua kromosom dari populasi secara acak, kemudian akan dipilih salah satu berdasarkan nilai fitnessnya. Biasanya yang sering digunakan adalah k = 2, tergantung dari jumlah kromosom yang tedapat di dalam populasi.

2.2.3.2 Perkawinan silang

Operator perkawinan silang digunakan untuk membentuk keturunan baru berdasarkan orangtua terpilih. Operator ini paling dominan dalam algoritma genetika dibandingkan dengan operator mutasi. Jumlah kromosom yang terlibat adalah 2 kromosom.

Perkawinan silang dilakukan dengan harapan kromosom-kromosom baru akan mempunyai bagian baik dari kromosom-kromosom lama dan tidak menutup kemungkinan menjadi kromosom-kromosom yang lebih baik Berikut ini akan dibahas beberapa jenis variasi perkawinan silang dalam algoritma genetika sehubungan dengan jenis pengkodean kromosom yang digunakan.

2.2.3.2.1 Perkawinan silang untuk pengkodean biner

Pengkodean biner merupakan jenis pengkodean yang paling sederhana sehingga banyak variasi operasi perkawinan silang dapat dilakukan. Beberapa jenis perkawinan silang tersebut adalah :

2.2.3.2.1.1 Perkawinan silang 1-titik

Proses perkawinan silang 1-titik dimulai dengan pemilihan 1 titik pada barisan bit kromosom secara acak sebagai titik perkawinan silang. Kromosom baru akan dibentuk dengan cara menyalin barisan bit orangtua pertama dari bit pertama sampai titik perkawinan silang, sedangkan sisanya disalin dari orangtua kedua.

Tabel 2.7 Contoh perkawinan silang 1-titik pada pengkodean biner

Kromosom orangtua 1	**11001**011
Kromosom orangtua 2	11011**111**
Keturunan	**11001111**

2.2.3.2.1.2 Perkawinan silang 2-titik

Proses perkawinan silang 2-titik diawali dengan pemilihan 2 titik perkawinan silang dari barisan bit kromosom secara acak. Kromosom baru akan dibentuk dengan cara menyalin barisan bit kromosom orangtua pertama dari bit pertama sampai dengan titik perkawinan silang pertama dan dari titik perkawinan silang kedua sampai dengan bit terakhir, sedangkan sisanya, yaitu dari titik perkawinan silang pertama sampai titik perkawinan silang kedua disalin dari orangtua kedua.

Tabel 2.8 Contoh perkawinan silang 2-titik pada pengkodean biner

Kromosom orangtua 1	**11**001011
Kromosom orangtua 2	11**011**11
Keturunan	**11011111**

2.2.3.2.1.3 Perkawinan silang seragam

Perkawinan silang seragam menghasilkan kromosom baru dengan cara menyalin bit-bit secara acak dari kedua orangtuanya.

Tabel 2.9 Contoh perkawinan silang seragam pada pengkodean biner

Kromosom orangtua 1	1**10**0101**1**
Kromosom orangtua 2	**1**10**111**1
Keturunan	**11011111**

2.2.3.2.1.4 Perkawinan silang aritmatik

Kromosom baru dihasilkan dari operasi aritmatika yang dilakukan terhadap kedua orangtuanya.

Tabel 2.10 Contoh perkawinan silang aritmatika pada pengkodean biner

Kromosom orangtua 1	11001011
Kromosom orangtua 2	11011111
Keturunan (AND)	**11001011**

2.2.3.2.1.5 Perkawinan silang rekombinasi

Versi paling sederhana dari rekombinasi adalah memilih secara acak dari semua nilai gen yang terdapat pada kedua orangtua untuk setiap gen pada anak. Berikut adalah contoh dari perkawinan silang rekombinasi:

Tabel 2.11 Contoh perkawinan silang rekombinasi pada pengkodean biner

Kromosom orangtua 1	15 9 12 14
Kromosom orangtua 2	13 4 16 2
Keturunan 1	15 4 16 14
Keturunan 2	14 9 12 1

2.2.3.2.2 Perkawinan silang untuk pengkodean permutasi

Pada pengkodean permutasi, jenis perkawinan silang yang sering dipergunakan adalah perkawinan silang 1-titik karena kesederhanaan prosesnya dapat menjaga konsistensi urutan nilai pada kromosom. Jenis perkawinan silang yang terdapat pada pengkodean biner tidak dapat dipergunakan pada pengkodean ini karena tidak dapat menjamin terjaganya konsistensi urutan nilai pada kromosom.

Proses perkawinan silang 1-titik dimulai dengan pemilihan satu titik perkawinan silang. Dari permutasi pertama sampai dengan titik perkawinan silang disalin dari orang tua pertama, sedangkan sisanya didapatkan dengan cara melihat satu persatu nilai pada orang tua kedua, jika belum ada pada kromosom keturunan, maka nilai tersebut ditambahkan.

Tabel 2.12 Contoh perkawinan silang 1-titik pada pengkodean permutasi

Kromosom orangtua 1	**1 2 3 4 5** 6 7 8 9
Kromosom orangtua 2	**4 5 3 6 8** 9 7 2 1
Keturunan (AND)	**1 2 3 4 5** 6 8 7 9

2.2.3.2.3 Perkawinan silang untuk pengkodean nilai

Semua metode perkawinan silang yang terdapat pada pengkodean biner dapat diterapkan pada pengkodean nilai karena tidak perlu memperhatikan urutan seperti ada pengkodean permutasi. Berikut adalah contoh perkawinan silang pada pengkodean nilai:

Tabel 2.13 Contoh perkawinan silang 1-titik pada pengkodean nilai bilangan bulat

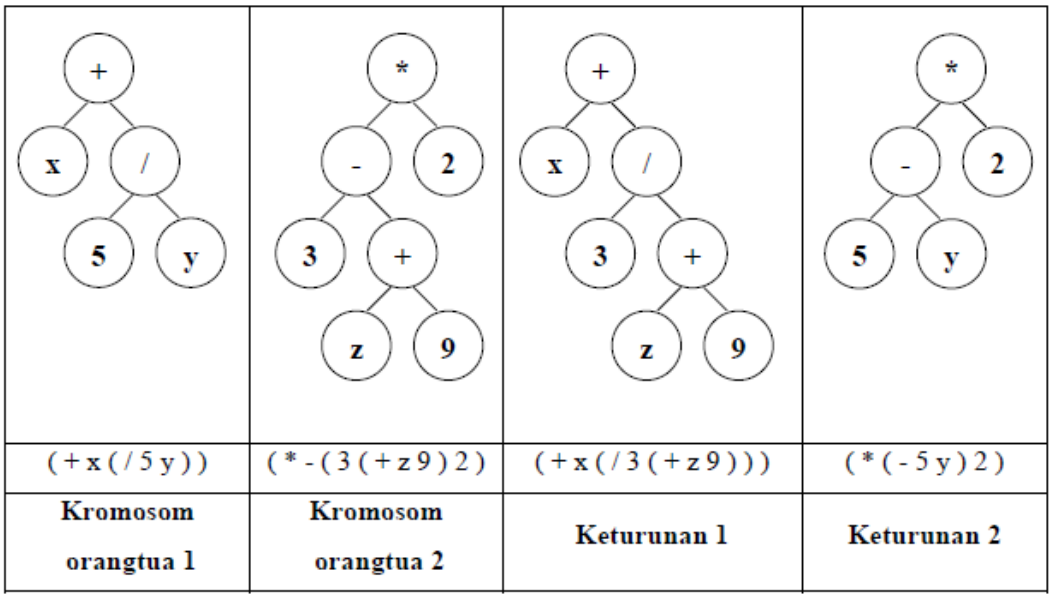
Kromosom orangtua 1	**10 39 45 12 89 34** 1 3 90 48 23 72 25 37
Kromosom orangtua 2	43 13 2 66 49 56 **10 89 53 20 75 99 4 8**
Keturunan	**10 39 45 12 89 34 10 89 53 20 75 99 4 8**

Tabel 2.14 Contoh perkawinan silang 2-titik pada pengkodean nilai bilangan bulat

Kromosom orangtua 1	**10 39 45** 12 89 34 1 3 **90 48 23 72 25 37**
Kromosom orangtua 2	43 13 2 **66 49 56 10 89** 53 20 75 99 4 8
Keturunan	**10 39 45 66 49 56 10 89 90 48 23 72 25 37**

2.2.3.2.4 Perkawinan silang untuk pengkodean pohon

Proses perkawinan silang pada pengkodean pohon dimulai dengan pemilihan satu titik perkawinan silang pada kedua orangtua, kemudian bagian dari orangtua yang berada di bawah titik perkawinan silang dipertukarkan untuk menghasilkan anak baru.



Gambar 2.4 Perkawinan silang untuk pengkodean pohon

2.2.3.3 Mutasi

Mutasi merupakan proses mengubah nilai dari 1 atau beberapa gen dalam suatu kromosom. Tujuannya adalah untuk membantu mempercepat terjadinya perbedaan diantara semua kromosom dalam populasi sehingga pencarian dapat menjelajah keseluruh ruang pencarian, tetapi mutasi tidak boleh terlalu sering dilakukan karena akan membuat algoritma genetika berubah menjadi pencarian acak.

Proses mutasi dalam biologi adalah mengubah isi allele gen pada suatu locus dengan nilai allele yang lain. Proses mutasi dilakukan secara acak, sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan fitness yang lebih baik dibandingkan dengan sebelum mutasi. Oleh karena itu terjadi kontraversi dalam penerapannya pada algoritma genetika. Mutasi seringkali tetap dipergunakan hanya saja dengan probabilitas mutasi yang kecil. Tentang probabilitas perkawinan silang dan mutasi akan dijelaskan pada bagian berikut pada bab ini.

Operator mutasi merupakan operasi pada suatu kromosom tertentu. Berikut adalah beberapa cara mutasi yang sering diterapkan dalam algoritma genetika menurut jenis pengkodean kromosomnya :

2.2.3.3.1 Mutasi pada pengkodean biner

Mutasi dalam pengkodean biner merupakan operasi yang sangat sederhana. Bit pada kromosom yang terpilih secara acak (atau menggunakan skema tertentu) diinversi nilainya, diubah nilainya menjadi lawannya.

Tabel 2.15 Contoh mutasi pada pengkodean biner

Kromosom sebelum mutasi	1 0 0 1 0 **1** 1 1
Kromosom sesudah mutasi	1 0 0 1 0 **0** 1 1

2.2.3.3.2 Mutasi pada pengkodean permutasi

Proses mutasi pada pengkodean biner tidak dapat diterapkan pada pengkodean permutasi karena konsistensi urutan permutasi harus selalu diperhatikan. Salah satu cara yang dapat dilakukan adalah dengan memilih dua posisi (locus) pada kromosom dan kemudian saling dipertukarkan

Tabel 2.16 Contoh mutasi pada pengkodean permutasi

Kromosom sebelum mutasi	1 **2** 3 4 5 6 **8** 9 7
Kromosom sesudah mutasi	1 **8** 3 4 5 6 **2** 9 7

2.2.3.3.3 Mutasi pada pengkodean nilai

Proses mutasi pada pengkodean biner dapat diterapkan pada pengkodean nilai karena tidak perlu menjaga konsistensi adanya urutan pada kromosom. Tetapi yang dilakukan bukanlah menginversi bit, melainkan tergantung dari jenis nilai yang dipergunakan. Misalnya nilai yang berupa bilangan riil dimutasi dengan cara saling menukar nilai dua gen yang terdapata pada kromosom, tetapi cara ini tidak memberikan manfaat untuk menjamin adanya perbedaan dalam populasi, semua kromosom dengan mudah memiliki nilai yang sama dan akhirnya terjadi konvergensi prematur. Cara lain yang lebih baik adalah dengan memilih

sembarang posisi gen pada kromosom, kemudian ditambahkan atau dikurangkan dengan suatu nilai kecil tertentu yang diambil secara acak. Cara ini juga berlaku pengkodean dengan bilangan bulat. Selain itu cara mutasi lain yang relevan dapat digunakan.

Tabel 2.17 Contoh mutasi pada pengkodean nilai riil dengan nilai yang ditambahkan atau dikurangkan dengan 0,10

Kromosom sebelum mutasi	1,43 **1,09** 4,51 **9,11** 6,94
Kromosom sesudah mutasi	1,43 **1,19** 4,51 **9,01** 6,94

2.2.3.3.4 Mutasi pada pengkodean pohon

Mutasi dapat dilakukan antara lain dengan mengubah operator (+, -, *, /) atau nilai yang terkandung pada suatu verteks pohon yang dipilih. Atau dapat juga dilakukan pemilihan dua verteks dari pohon dan saling mempertukarkan operator atau nilainya.

2.2.4 Update Generasi

Setelah proses seleksi, perkawinan silang dan mutasi yang dilakukan terhadap suatu populasi, proses berikutnya adalah pemilihan kromosom untuk membentuk generasi baru yang disebut update generasi. Dalam algoritma genetik Holland, proses update yang dilakukan adalah update secara generasi (generational update), yaitu menggantikan keseluruhan kromosom pada generasi sebelumnya dengan kromosom-kromosom yang didapat dari hasil seleksi, perkawinan silang dan mutasi. Penekanan selektif pada skema update ini seluruhnya berasal dari orangtua.

Selain itu terdapat skema update secra kontinu (continuous update) yang mengizinkan orangtua dan anak untuk bercampur dalam satu generasi. Mulanya dipilih k orangtua secara acak, kemudian dibuat anak sebanyak k dengan memilih orangtua tersebut secara seragam. Kromosom pada populasi baru merupakan pemilihan kromosom antara orangtua terpilih dan anak yang dihasilkan berdasarkan fitnessnya. Dengan cara ini penekanan selektif berasal seluruhnya dari pemilihan terhadap yang bertahan hidup (survivor).

Skema update lainnya adalah steady-state update, dimana dilakukan pemilihan dua orangtua dari populasi yang kemudian digunakan untuk membentuk 1 (atau 2) anak. Hasilnya digunakan untuk menggantikan :

1. Orangtuanya
2. Kromosom terjelek dalam populasi
3. Kromosom tertua dalam populasi

Suatu strategi bantuan yang baik untuk digunakan dengan skema continous update adalah elitism, yang berarti mempertahankan kromosom-kromosom baik dalam populasi lama dengan menyertakannya lagi pada populasi baru. Sejumlah kromosom terbaik pada populasi lama dapat hilang karena proses perkawinan silang dan mutasi. Secara teori, elitism dapat meningkatkan kemampuan dari algoritma genetika karena mempertahankan kromosom yang baik dari populasi lama, tetapi kadangkala hal itu justru dapat menyebabkan konvergensi prematur karena nilai fitness terjebak pada optimum lokal.

2.2.5 Parameter dalam Algoritma Genetika

Dua parameter dasar yang penting dari algoritma genetika adalah probabilitas perkawinan silang dan probabilitas mutasi.

Probabilitas perkawinan silang menyatakan seberapa sering proses perkawinan silang akan terjadi diantara 2 kromosom orangtua. Jika tidak terjadi perkawinan silang, keturunan merupakan salinan mutlak dari orangtuanya. Jika terjadi perkawinan silang, keturunan dibuat dari bagian-bagian kromosom orangtua. Jika probabilitas perkawinan silang 100%, maka keseluruhan keturunan dibuat dengan perkawinan silang. Jika probabilitas perkawinan silang 0%, maka seluruh generasi baru dibuat dari salinan kromosom-kromosom dari populasi lama, tetapi ini tidak berarti bahwa generasi baru sama dengan yang lama karena adanya penekanan selektif. Meskipun perkawinan silang bertujuan untuk mendapatkan kromosom yang memiliki bagian baik dari orangtuanya atau bahkan menjadi lebih baik dari orangtuanya, ada baiknya juga jika kita membiarkan beberapa bagian dari populasi untuk bertahan ke generasi berikutnya.

Dari hasil penelitian yang sudah dilakukan oleh praktisi algoritma genetika terbukti bahwa angka probabilitas perkawinan silang sebaiknya cukup tinggi, yaitu antara 80% sampai 95% untuk memberikan hasil yang baik. Untuk beberapa masalah tertentu probabilitas perkawinan silang 60% memeberikan hasil yang lebih baik (Obitko, 1998).

Probabilitas mutasi menyatakan seberapa sering bagian-bagian kromosom akan dimutasikan. Jika tidak ada mutasi, keturunan diambil/ disalin langsung setelah perkawinan silang tanpa perubahan. Jika mutasi dilakukan, bagian-bagian keromosom diubah. Jika probabilitas mutasi 100%, semua kromosom diubah. Jika

probabilitas mutasi 0%, tidak ada yang diubah. Probabilitas mutasi dalan algoritma genetika seharusnya diberi nilai yang kecil. Umumnya probabilitas mutasi diset untuk mendapatkan rata-rata satu mutasi per kromosom, yaitu angka/ allele = 1/ (panjang kromosom). Dari hasil yang sudah pernah dicoba ternyata angka probabilitas terbaik adalah antara 0,5% sampai 1%. Mutasi diperlukan untuk menjaga perbedaan kromosom dalam populasi, untuk menghindari terjadinya konvergensi prematur.

Parameter lain yang juga ikut menentukan efisiensi kinerja algoritma genetika adalah ukuran populasi yaitu banyakknya kromosom dalam satu populasi atau satu generasi. Jika terlalu sedikit kromosom dalam populasi, algoritma genetika mempunyai kemungkinan yang sedikit untuk melakukan perkawinan silang dan hanya sebagian kecil dari ruang pencarian yang dieksplorasi. Sebaliknya, jika terlalu banyak jumlah kromosom, algoritma genetika cenderung lambat dalam menemukan solusi. Ukuran populasi yang sering digunakan oleh peneliti yang sudah ada adalah antara 20 sampai 30, tetapi kadang ukuran 50 sampai 100 dilaporkan baik. Beberapa penelitian menujukkan bahwa menambah jumlah populasi tidak akan terlalu berguna karena tidak mempercepat penyelesaian masalah. Riset juga menunjukkan bahwa ukuran populasi yang terbaik ditentukan dari jenis pengkodean, artinya, jika terdapat ukuran kromosom 32 bit, ukuran populasi seharusnya juga 32, begitu pula jika ukuran kromosom 16 bit, maka ukuran populasi adalah 16 (Obitko, 1998).

Penekanan selektif dilakukan dengan memilih jenis seleksi yang sesuai. Umumnya jenis seleksi roda roulette sering digunakan, tetapi kadang seleksi rangking dilaporkan lebih baik. Terdapat juga beberapa metode lebih lanjut yang dapat mengubah parameter seleksi yang digunakan selama ekseskusi algoritma genetika. Prinsip elitism disarankan untuk digunakan jika tidak digunakan metode lain untuk menyimpan solusi terbaik pada suatu populasi ke populasi berikutnya. Sebaikya juga dicoba seleksi steady state untuk proses update populasi.

Terakhir adalah jenis pengkodean, perkawinan silang dan mutasi. Jenis pengkodean dipilih berdasarkan pada masalah dan juga ukuran instance-nya. Begitu pula untuk pemilihan jenis operator perkawinan silang dan mutasi.

2.2.6 Pengenalan Delphi

Delphi merupaka salah satu bahasa pemrograman tingkat tinggi erbasis *Windows*. Delphi digolongkan ke dalam bahasa pemrograman visual yang menitik beratkan pada pemrograman berorientasi objek *object oriented programming*. Delphi dikembangkan menggunakan bahsa pascal. 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.*

2.2.6.1 Keunggulan Delphi

1. IDE ***(****Integrated Development Environment****)*** atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program,
2. Proses Kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada Delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
3. Mudah digunakan, source kode delphi yang merupakan turunan dari pascal, sehingga tidak diperlukan suatu penyesuain lagi.
4. Bersifat multi purphase, artinya bahasa pemograman Delphi dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi

2.2.6.2 OOP (Object Oriented Programming)

OOP adalah metode pemograman dengan membentu sebuah aplikasi yang mendekati keadaan dunia yang sesungguhnya. Hal itu bisa dilakukan dengan cara mendisain *object* untuk menyelesaikan masalah.

2.2.6.2.1 Tiga unsure OOP

1. *Encapsulation* atau pemodelan

Encapsulation adalah konsep penggabungan data dengan operator. Dalam konsep pemodelan data dan operasi menjadi satu kesatuan yang disebut object.

Encapsulation juga disebut dengan penyembunyian informasi *(information hiding)* . Contoh = ketika kita menyalakan sebuah TV kita tidak tahu apa yang terjadi dengan proses dan percakapan antar alat yang berhubungan dengan TV tersebut sehingga mampu menampilkan sebuah gambar. Atau = ketika melakukan klik pada subuah object dalam suatu aplikasi program kita tidak tahu apa yang dilakukan program sehingga ditampilkan hasil yang sesuai. Catatan = dari dua contoh kasus tersebut terdapat kesamaan proses mengenai *information hiding* yang tidak diketahui oleh user sampai hasil ditampilkan.

2. *Inheritance* atau penurunan

Inherintance adalah sebuah object yang dapat diturunkan menjadi object yang baru dengan tidak menghilangkan sifat asli dari object tersebut. Contoh = TV merupakan salah satu media elektronik yang dgunakan untuk menampilkan gambar dengan tujuan untuk memberikan informasi kepada konsumen. Secara umum TV memunyai cara kerja yang sama dengan media elektronik yang lain dlam proses penyampaian informasi, tetapi mempunyai sifat yang unik yang dapat membedakan dengan media electronik yang ada. Atau = Dalam aplikasi program kita sering menggunakan command button, untuk beberapa perintah yang berbeda.

3. *Polymorphism* atau Polimorfisme

Polymorphism merupakan penggunaan berbagai macam object yang berbeda tetapi secara fungsi bergantung pada satu object sebagai induk, dengan cara pelaksanaan yang berbeda – beda. Contoh = TV dan radio adalah media electronik yang mempunyai sistem yang sama tentang bagaimana meyebarkan suatu

informasi, tetapi cara kerja dari masing – masing system pasti berbeda. Atau Object Simpan dan Update adalah icon yang berasal dari induk yang sama yaitu , command button tetapi cara kerja tersebut berbeda – beda.

2.2.7 Pengertian Database

Database dalam pembuatan program dengan bahasa pemrograman berbasis visual, merupakan suatu hal yang sangat penting. Karena database sering digunakan untuk menampung keluar masuknya data dalam suatu program aplikasi. Terkadang database juga sering digunakan untuk membantu proses pengolahan data yang rumit.

Bahasa pemrograman Delphi dilengkapi dengan beberapa *tool* yang sangat membantu dalam proses pembuatan program. Di antaranya adalah tool dengan nama database desktop. Database yang bias digunakan untuk membuat program menggunakan Delphi ada berbagai macam jenis, di antaranya adalah database dengan tipe paradox, MS Acces, FoxPro, Dbase, Oracle, MS SQL Server dan Interbase.

2.2.8 Alat Pengembang Sistem

2.2.8.1 Diagram konteks

Diagram konteks menggambarkan aplikasi dalam satu lingkaran dan hubungan dengan entitas luar. Dimana lingkaran tersebut menggambarkan keseluruhan proses dalam aplikasi. Dalam penggambaran ini, sistem dianggap sebagai sebuah objek yang tidak dijelaskan secara rinci, karena yang ditekankan adalah interaksi sistem dengan lingkungan yang mengaksesnya.

2.2.8.2 DFD (*Data Flow Diagram*)

Data Flow Diagram adalah diagram sistem yang menggambarkan cara kerja aplikasi secara *logic*. Mulai dari tingkat paling tinggi sampai dengan tingkat paling rendah. Pada perancangan ini terdiri dari perancangan awal *(preliminary design*) dan perancangan rinci *(detailed design)* sesuai dengan tahap-tahap rekayasa perangkat lunak. Adapun penjelasan dari perancangan awal adalah perancangan sistem yang menggambarkan tentang hubungan antara sistem dengan lingkungan luar sistem.

Hubungan ini dapat digambarkan dengan menggunakan diagram konteks. Sedangkan perancangan rinci adalah perancangan sistem yang menggambarkan tentang proses yang terjadi pada sistem serta arus data yang mengalir antar proses. DFD merupakan alat yang digunakan pada metode pengembangan sistem yang terstruktur, DFD ini menggambarkan arus data di dalam sistem yang terstruktur dan jelas serta merupakan dokumentasi sistem yang baik.

2.2.8.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu model diagram yang menyatakan keterhubungan suatu entity dengan entity yang lain. Atau juga dapat dikatakan sebagai sebuah teknik untuk menggambarkan informasi yang dibutuhkan dalam sistem dan hubungan antar data-data tersebut.

Secara terjemahan dalam bahasa Indonesia, *Entity Relationship Diagram* adalah diagram relasi atau keterhubungan entitas. Dari model *Entity Relationship*

Diagram akan didapatkan data-data yang dibutuhkan sistem. Dengan begitu maka akan didapatkan pula kejelasan aktivitas yang dilakukan dalam sistem.

Didalam *Entity Relationship Diagram* (ERD) dikenal beberapa komponen, yaitu sebagai berikut :

1. Entitas (*Entity*)

   Adalah suatu objek yang memiliki hubungan dengan objek lain. Dalam ERD digambarkan dengan bentuk persegi panjang.

2. Hubungan (*Relationship*)

   Dimana entitas dapat berhubungan dengan entitas lain, hubungan ini disebut dengan *entity relationship* yang digambarkan dengan garis. Berikut adalah beberapa jenis hubungan yang dapat terjadi dalam suatu basis data :

   1) Relasi 1-1 (*One to One*) Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu.

   2) Relasi 1-N (*One to Many*) Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak, tetapi tidak sebaliknya, dimana *file* kedua hanya berhubungan dengan satu entitas pada *file* pertama.

   3) Relasi N-1 (*Many to One*) Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding satu, tetapi tidak sebaliknya,

dimana *file* pertama hanya berhubungan dengan satu entitas pada *file* kedua.

4) Relasi N-N (*Many to Many)* Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak.

3. Atribut

Atribut adalah elemen dari entitas yang berfungsi sebagai deskripsi karakter entitas dan digambarkan dengan bentuk elips.

2.2.8.4 Kamus Data

Kamus data atau data *directory* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap perancangan sistem, kamus data dapat digunakan untuk merancang *input, output,* dan merancang *database* program. Kamus data dibuat berdasarkan arus data yang ada.