BAB 2
TINJAUAN PUSTAKA

2.1 Sejarah Perusahaan

Di Indonesia pelistrikan dimulai sejak awal abad ke-19, digunakan untuk penerangan sebagai ganti lampu-lampu minyak dan lampu gas. Pada saat itu perusahaan pelistrikan di Indonesia dipegang dan diselenggarakan secara monopoli oleh perusahaan swasta belanda. Pada awal tahun 1950, saat Indonesia masih berada pada masa penjajahan oleh pemerintah hindia belanda, perusahaan listrik di daerah bandung mulai berdiri. Perusahaan listrik tersebut di monopoli oleh perusahaan partikelir belanda yaitu "Bandoengsche Electictiet Maatschappij"(BEM), yang memperoleh izin dari pemerintah Hindia Belanda untuk mendirikan perusahaan listrik di daerah jajahannya dengan tugas membuat jaringan-jaringan listrik di kota bandung dan sekitarnya.

Sedangkan pada tahun 1913, pengelola usaha pembangkit dan penyalur tenaga listrik dilaksanakan oleh "Land Waterkraccht Bedriif" (LWB) yang mempunyai wilayah kerja diseluruh Indonesia. Kemudian pada tahun 1919, perusahaan BEM dihapuskan dan digantikan oleh perusahaan berbadan hukum dan berbentuk perseroan terbatas yang di didalamnya yang terdapat 24 sero (saham) dan tugas melaksanakan pengusahaan distribusi tenaga listrik di daerah Bandung dan sekitarnya. Perusahaan tersebut diberi nama GEBEO NV (Gemeanschpplij Electricteit Bedrijf Voor Bandoeng Omstroken) dan prailihan itu tepatnya dilaksanakan pada tanggal 1 Januari 1920.

Seiring dengan kekalahan Belanda rerhadap Jepang, maka pada tahun 1942 sampai dengan tahun 1945 penguasaan dan pengusahaan pelistrikan di Indonesia pada umumnya dan jawa barat pada khususnya secara otomatis beralih dari pemerintah Belanda ke Pemerintah Jepang. Pada masa penjajahan Jepang pengusahaan dan distribusi listrik dilakukan oleh "Djawa Denki Djigyo Sha Bandoeng Shi Sah" yang bertugaas mengelolapembangkit tenaga listrik dan penyalurannya pada kurun waktu 1942 – 1943, kemudian berubah menjadi "Denki Kosha" dengan wilayah kerja seluruh wilayah pulau jawa.

Sehubungan dengan didudukinya Indonesia oleh belanda dan pindahnya pemerintah Indonesia ke Yogyakarta pada tahun 1948, maka pengusahaan dan pendistribusian tenega listrik khususnya di jawa barat dan Jakarta dikuasai kembali oleh perusahaan belanda yaitu GEBEO NV, sedangkan untuk pembangkit dan penyalurannya dilakukan oleh pemerintah Indonesia yaitu perusahaan Negara untuk pembangkit tenaga listrik (PENUPETEL) dengan wilayah kerja diseluruh jawa barat dan Jakarta. Pada tahun 1952 berdiri pula perusahaan Negara untuk pendistribusian tenaga listrik (PENUDITEL) dan akhirnya PENUPETEL dan PENUDITEL berada di bawah naungan jawatan listrik.

Berdasarkan keputusan presiden (Kepres) No.163 tanggal 3 oktober 1953 tentang Nasionalisasi peruhaan listrik milik bangsa Belanda, yaitu jika konsesi perusahaannya telah berakhir,maka beberapa perusahaan listrik swasta tersebut diambil alih dan digabungkan ke jawatan tenaga. Kemudian pada tahun 1958, Dewan Perwakilan Rakyat (DPR) dan pemerintah RI menerbitkan undang - undang tentang nasionalisasi semua perusahaan belanda dan peraturan pemerintah republik Indonesia (PP) No.18 tahun 1959 tentang nasionalisasi perusahaan listrik dan gas milik Belanda. Berdasarkan peraturan listrik dan gas (P3LG) dikenai proses pengalihan kepemilikan.

Melalui surat menteri pekerjaan umum dan tenaga No.P25/45/14 tanggal 23 september 1958, jawatan tenaga diubah menjadi perusahaan listrik Negara. Setelah dewan direktur perusahaan listrik Negara (DD PLN) terbentuk, yaitu pada tahun 1959 P3LG dibubarkan. Berdasarkan Undang - Undang No.19 tahun 1960 tentang perusahaan Negara dan peraturan pemerintah RI No.67 tahun 1961 badan pimpinan umum perusahaan listrik Negara (BPU PLN) terbentuk. PLN terbentuk melalui PP No.19 tahun 1965 yang bernaung dibawah departemen pekerjaan umum dan tenaga listrik (PUTL). Menteri pekejaan umum dan tenaga listrik pada saat itu menerbitkan surat keputusan menteri pekerjaan umum dan tenaga listrik No.013/PRT/1957 tanggal 8 september 1957 tentang organisasi dan tata kerja perusahaan umum listrik Negara. Bedasarkan peraturan menteri pekerjaan umum dan tenaga listrik No.1/PRT/1965 tanggal 21 januari 1965 dan PP No.19 tahun 1965, diadakan reorganisasi PLN dimana BPU PLN dihapus dan ditingkat daerah

dibentuksusunan organisasi PLN yang disebut PLN Exploitasi XI, yang berkedudukan di bandung dengan wilayah kerja seluruh jawa barat kecuali Tanggerang, Bogor, Sukabumi, dan Banten.

Perkembangan selanjutnya adalah dikeluarkannya PP No.18 tahun 1972 tentang pembentukan perusahaan umum listrik Negara yang berkantor pusat di Jakarta dengan peraturan menteri pekerjaan umum dan tenaga listrik (PUTL) No.043/PMS/1975 tanggal 9 september 1975 tenteng tata kerja perusahaan – perusahaan listrik Negara, maka PLN mengadakan reorganisasi dalam tugas dan wilayah daerah. Kemudaian berdasarkan pengumuman No.05/D.III/Sek/1975 tanggal 14 juli 1975 maka PLN Ekploitasi XI diganti namanya menjadi perusahaan listrik Negara distribusi III yang bertugas untuk melaksanakan dan mangusahakan distribusi listrik langsung kepada pemakai atau konsumen.

Perusahaan listrik Negara yang didirikan dengan PP No.19 tahun 1956 Jo, PP No.11 tahun 1956 dan PP No.30 tahun 1970 ditegaskan statusnya menjadi suatu perusahaan umum (PERUM). Sebagaimana termasuk dalam pasal 2 ayat (2) UU No.9 tahun 1969 dengan nama perusahaan listrik Negara. Dengan adanya PP RI No.23 tahun 1994 pada tanggal 16 juli 1994 tentang pengalihan bentuk perusahaan umum (PERUM) listrik Negara menjadi perseroan terbatas (PT) yang termuat dalam lembaran Negara RI tahun 1994 No.34 maka bentuk perusahaan listrik distribusi jawa barat berubah menjadi PT. PLN (PERSERO) distribusi jawa barat sejak tanggal 30 juli 1994 sesuai akte pendirian kemudian melalui surat keputusan direksi PT. PLN (PERSERO) No,28.k/010/DIR/2001 tanggal 20 februari 2001, PT (Persero) distribusi ajwa barat di ubah menjadi PT.PLN (Persero) unit bisnis distribusi jawa barat. Kemudian melalui surat keputusan PT.PLN (Persero) No.120.K/010/DIR/2002 tanggal 27 agustus 2002, PT.PLN (Persero) unit distribusi jawa barat diubah menjadi PT.PLN (Persero) istribusi Jawa barat dan Banten.

Visi, Misi dan Motto PT. PLN (Persero)

Visi PT. PLN (Persero) Distribusi Jawa Barat dan Banten Area Bandung Rayon Bandung Timur adalah “Diakui Perusahaan kelas Dunia yang bertumbuh - kembang, unggul, dan Terpercaya dengan bertumbuk pada potensi insansi”.

Visi PT. PLN (persero) Distribusi Jawa Barat dan Banten Area Bandung Rayon Bandung Timur tersebut diuraikan satu per satu diantaranya:

Visi PT. PLN (Persero) Distribusi Jawa Barat dan Banten Area Bandung Rayon Bandung timur, diakui :

Mencerminkan cita – cita untuk meraih pengakuan dari pihak luar yang menunjukan bahwa PT. PLN pantas dipandang sebagai perusahaan kelas dunia.

Visi PT. PLN (Persero) Distribusi Jawa Barat dan Banten Area Bandung Rayon Bandung Timur, Kelas dunia :

a. Menunjukan kinerja yang melebihi ekspektasi pihak - pihak yang berkepentingan.
b. Memberikan layanan yang mudah,terpadu, dan tuntas dalam berbagai kelistrikan.
c. Menjalin hubungan kemitraan yang akrab dan setara dengan pelanggan serta mitra usaha nasional dan internasional.
d. Diakui pelanggan dan mitra kerja sebagai perusahaan yang mampu memenuhi standar mutahir dan paling baik.

Visi PT. PLN (Persero) Distribusi Jawa Barat Dan Banten Area Bandung Rayon Bandung Timur, Bertumbuh Kembang :

a. Antisipatif terhadap perkembangan lingkungan usaha dan selalu siap menghadapi berbagai tantangan.
b. Secara konsisten menunjukan kinerja yang lebih baik.

Visi PT. PLN (Persero) Distribusi Jawa barat dan Banten Area Bandung Rayon Bandung Timur, Unggul :

a. Menjadi yang terbaik dalam bisnis kelistrikan dan memenuhi tolak ukur mutahir dan terbaik.
b. Mamposisikan diri sebagai perusahaan yang terkemuka dalam pencaturan bisnis kelistrikan dunia.
c. Mengelola usaha dalam mengedepankan pemberdayaan dan potensi insansi secara maksimal.

d. Meningkatkan kualitas proses, system, produk dan pelayanan secara berkesinambungan.

Visi PT. PLN (Persero) Distribusi Jawa barat dan Banten Area Bandung Rayon Bandung Timur, Terpercaya :

a. Memegang teguh etika bisnis yang tertinggi.
b. Menghasilkan kinerja terbaik secara konsisten.
c. Menjadi perusahaan pilihan.

Visi PT. PLN (Persero) Distribusi Jawa barat dan Banten Area Bandung Rayon Bandung Timur, Potensi Instansi :

a. Keberhasilan perusahaan lebih ditentukan oleh kesadaran anggota perusahaan untuk memunculkan seluruh potensi mereka dalam wujud wawasan aspiratif, motivasi kerja, semangat belajar inopatif dan semangat bekerja sama.
b. Potensi insansi diperkaya dengan kompetensi yang terbentuk dari segi pengetahuan kontekstual, keterampilan, kemampuan, pengalaman dan jaringan strategi korporat.

Misi PT. PLN (Persero) Distribusi Jawa barat dan Banten Area Bandung Rayon Bandung Timur :

1. Menjalankan bisnis kelistrikan dan bidang lain yang terkait, berorientasi kepada kepuasan pelanggan, anggota perusahaan , dan pemegang saham.
2. Menjadikan tenaga listrik sebagai media untuk meningkatkan kualitas kehidupan masyarakat.
3. Mengupayakan agar tenaga listrik menjadi pendorong kegiatan ekonomi.
4. Menjalankan kegiatan usaha yang berwawasan lingkungan.

Demi mewujudkan visi dan misi perusahaan, maka landasan filosofi PT. PLN (Persero) Distribusi Jawa Barat dan Banten Area Bandung Rayon Bandung Timur adalah mempunyai komitmen yang tinggi terhadap kepentingan pelanggan dengan mewujudkan sumber daya manusia sebagai sumber daya penting perusahaan.

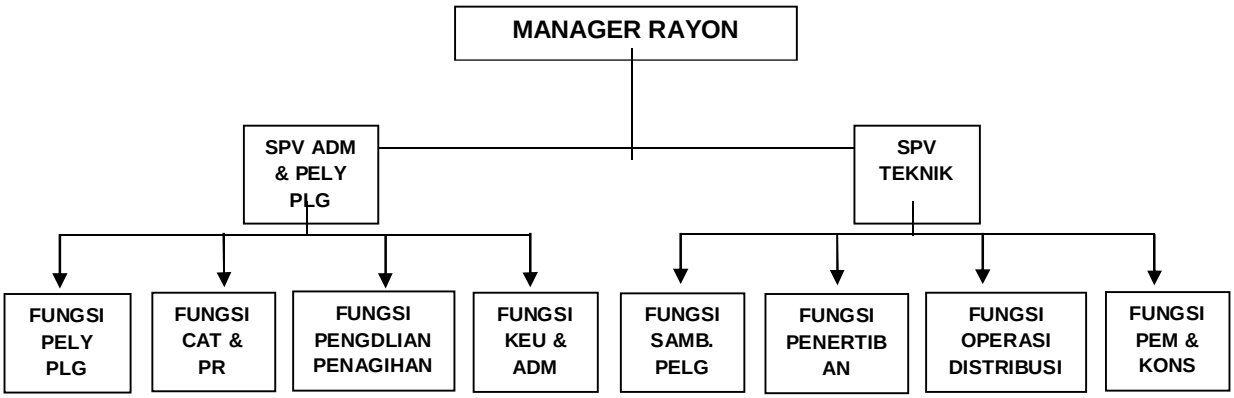
2.1.1 Struktur Organisasi Perusahaan

Struktur Organisasi merupakan suatu kerangka kerja yang memperhatikan adanya suatu pembagian tugas, wewenang dan tanggung jawab dari masing-masing bagian dalam melaksanakan aktivitas perusahaan. Struktur organisasi harus disusun secara teratur sehingga menggambarkan aktivitas perusahaan secara keseluruhan, mencakup susunan dan tugas masing-masing bagian, serta hubungan yang satu dengan yang lainnya.

Struktur Organisasi yang ada dalam PT PLN (Persero) Rayon Bandung Timur, berbentuk Organisasi Fungsi karena disusun atas dasar kegiatan dari tiap-tiap fungsi sesuai dengan kepentingan perusahaan, dimana tiap-tiap fungsi/kegiatan seolah-olah terpisah berdasarkan atas bidang keahliannya. Sekalipun demikian tiap-tiap fungsi/kegiatan tidak dapat berdiri sendiri, karena antara satu fungsi dengan fungsi yang lainnya saling ketergantungan.

Dilihat dari bentuk organisasi, maka wewenang tertinggi dalam PT PLN (Persero) Rayon Bandung Timur berada pada Manajer. Manager merupakan pimpinan utama yang dibantu oleh para Supervisor dan bagian-bagiannya. Susunan organisasi PT PLN (Persero) Rayon Bandung Timur adalah sebagai berikut:

STUKTUR ORGANISASI
PT PLN (Persero) DISTRIBUSI JAWA BARAT DAN BANTEN
AREA DAN PELAYANAN BANDUNG
RAYON BANDUNG TIMUR



Gambar 2.1 Struktur Organisasi

Sumber : *PT. PLN (Persero) Distribusi Jawa Barat dan Banten Area Bandung Rayon Bandung Timur 2012*

2.1.2. Uraian Tugas Perusahaan

Adapun uraian tugas dari masing-masing bagian adalah sebagai berikut :

1. Manajer Rayon

a. Memenuhi target bidang pemasaran
b. Memenuhi target bidang efesiensi
c. Memenuhi target mutu dan keandalan
d. Memenuhi target pelayanan
e. Memenuhi target kualitas SDM

2. Fungsi Pelayanan Pelanggan

a. Monitoring pelayanan pelanggan penyiapan SIP dan SP JBTL 1 & 3 Phase PB/TD
b. Monitoring pelayanan pelanggan untuk penyiapan dan pengarsipan khusus SP JBTL PB/TD > 82,5 kVA

c. Monitoring menyiapkan data pengukuran dan survey untuk pelanggan PB/TD Phase dan uji petik survey 1 Phase
d. Mengkoordinir pembuatan kajian inovasi produk unggulan PP & SAR
e. Mengkoordinir pelayanan pelanggan untuk melayani permohonan terif pesta, geser kWh, balik nama, dll
f. Monitoring UP dating data pelanggan pasang baru pada LAN (maf invo) & pemantauan mutasi pelanggan
g. Membuat perencanaan Forcacast penjualan tenaga listrik
h. Mengkoordinir Administrasi PP & SAR
i. Mengkoordinir pembenahan data induk pelayanan potensial dan implementasi fokus 100 dan fokus 5000
j. Melaksanakan promosi pemasaran PB, TD, program produk unggulan dan produk inovasi kantor Distribusi
k. Mengkoordinir penyiapan laporan bulanan
l. Mengkoordinir pemeliharaan aplikasi CIS

3. Fungsi Pencatatan Meter & Pengolahan Rekening

a. Menyusun dan merencanakan pembacaan meter
b. Melaksanakan pembacaan meter pelanggan potensial > 10.000 VA
c. Mengkoordinir pelaksanaan tugas pembacaan meter
d. Mempersiapkan dan mengevaluasi proses pemasukan data (data entry) PDE sampai cetak rekening

4. Fungsi Pengendalian Penagihan

a. Monitoring penjualan rekening keseluruhan (semua golongan)
b. Monitoring jumlah tunggakan rekening listrik
c. Monitoring semua laporan penagihan dan pengawasan kredit
d. Monitoring pengendalian rekening yang masuk UPR
e. Monitoring pelaksanaan TUL VI – 03
f. Montoring pelaksanaan penghapusan rekening UPR

g. Monitoring pelaksanaan penjualan rekening off line di payment point KUD dan Bank

5. Fungsi Keuangan dan Administrasi

a. Mengatur dan membina staf termasuk satpam yang berada dibawahnya dapat melaksanakan tugas sesuai bidang dan tanggung jawabnya masing-masing
b. Mengendalikan pelaksanaan pembuatan surat keluar, surat masuk, pencatatan surat-surat dokumentasi serta pendistribusiannya sesuai dengan ketentuan yang berlaku
c. Memeriksa permintaan barang, ATK, AKBC dan memeriksa bukti pembayaran baik kepada pegawai maupun kepada pihak ketiga
d. Mengendalikan penggunaan anggaran pengeluaran serta memonitor anggaran pendapatan agar sesuai dengan rencana yang telah ditetapkan
e. Pengawasan dan memeriksa buku kas harian, baik penerimaan maupun pengeluaran serta rekonsiliasi bank
f. Mengawasi dan mengendalikan berkas kepegawaian yang meliputi biodata pegawai, absensi, TKK dan gaji
g. Memonitor, memeriksa serta mengevaluasi laporan keuangan (laba/rugi, neraca, ratio keuangan, informasi tambahan)
h. Mengendalikan pengelolaan fasilitas gedung, rumah tangga agar dapat mendukung kelancaran tugas pegawai yang memerlukannya
i. Membantu penyelesaian masalah perusahaan yang berkaitan dengan hukum
j. Membina hubungan baik dengan publik internal dan eksternal perusahaan

6. Fungsi Sambungan Pelanggan

a. Pencapaian target penambahan pelanggan (Delta pelanggan) pertahun
b. Pencapaian target penambahan pelanggan (data kVA) pertahun

c. Menekan saldo tunggakan
d. Meningkatkan pelayanan terhadap pelanggan
e. Tertib Administrasi teknik dan percepatan pelaporan pencapaian target kVA tersambung pertahun
f. Tertib Administrasi teknik dalam pengadaan material
g. Menjaga keandalan sistem distribusi SRayon Bandung Timur

7. Fungsi Penertiban (P2TL)

a. Memonitor dan melaksanakan Penertiban pemakaian Tenaga Listrik (P2TL).
b. Memonitor dan melaksanakan pemeriksaan kWh macet (numpuk)
c. Memeriksa dan memonitor pemakaian listrik pelanggan yang tidak normal.
d. Melakukan koordinasi dengan Supervisor lainnya yang terkait dan bila diperlukan melakukan koordinasi dengan instansi/lembaga hukum.

8. Fungsi Operasi Distribusi

a. Memonitor pelayanan dan pelaksanaan mengatasi gangguan dari pelanggan yang terganggu aliran listrik tegangan rendah
b. Memonitor perbaikan jaringan-jaringan udara saluran rumah
c. Merehab/mengganti jaringan tegangan rendah yang terbuka SR/TR dari BC (tembaga) menjadi TIC (aluminium)/kabel yang terbugkus
d. Memonitor dan memperbaiki tegangan minimal dibawah 220 volt menjadi normal 220 volt
e. Memonitor dan penggantian tiang-tiang yang keropos
f. Membuat perencanaan perluasan jaringan distribusi
g. Mengusulkan penggantian dan penambahan bagi trafo-trafo gardu distribusi yang telah over load/kelebihan beban pemakaian.

9. Fungsi Pemeliharaan dan Konstruksi

a. Perluasan jaringan atau jurusan di rayon Bandung Timur.
b. Penertiban jaringan yang ada di rayon Bandung Timur.
c. Penggeseran tiang listrik yang sudah terpasang di area lokasi.
d. Penambahan gardu pada lokasi di rayon Bandung Timur.

2.1.3. Kegiatan PT. PLN (Persero)

PT PLN (Persero) adalah suatu perusahaan yang bertujuan untuk memenuhi kebutuhan dan tuntutan kebutuhan global di bidang ketenagalistrikan guna tercapainya tujuan pembangunan nasional.

Adapun aspek – aspek kegiatan PT. PLN (Persero) Distribusi Jawa Barat dan Banten Area Bandung Rayon Bandung Timur antara lain :

1. Produk

   Produk yang ditawarkan adalah jasa yang berupa pelayanan kepada pelanggan diantaranya : permintaan perubahan daya, perubahan golongan tarif, pemasangan baru,dsb.

2. Pasar

   Pasar terdiri dari semua pelanggan potensional yang sama – sama mempunyai kebutuhan atau keinginan yang mungkin ingin dan mampu terlibat dalam pertukaran untuk memuaskan kebutuhan dan keinginan.

   Adapun segmentasi diantaranya ;

   a. Segmentasi Geografis

   Yaitu pembagian pasar menjadi unit – unit geografis yang berbeda – beda pada PT. PLN (Persero) distribusi Jawa Barat dan banten Area Bandung Rayon Bndung Timur, batas pasar yang dilayani untuk wilayah:

   - Bagian Timur, hingga daerah Sukamiskin
   - Bagian Barat, hingga daerah Simpang lima
   - Bagian Utara, hingga daerah Awiligar, Cimeunyan
   - Bagian Selatan, hingga daerah Gatot Subroto

b. Segmentasi Demografis

- Untuk konsumen listrik yang bertegangan rendah, yaitu untuk kebutuhan rumah tinggal.
- Untuk konsumen listrik yang bertegangan sedang, yaitu untuk kebutuhan social seperti sekolah, tempat peribadatan, penerangan jalan,dll.
- Untuk konsumen listrik yang bertegangan tinggi yaitu untuk kebutuhan industri.

2.2. Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi. Beberapa teori yang terkait dengan pembangunan aplikasi adalah teori-teori mengenai pengertian algoritma genetika, pengertian individu, struktur umum algoritma genetika, dan komponen - komponen algoritma, serta membahas tentang Sistem Operasi Android, JDK, Eclipse, SQLite, UML dan JAVA.

2.2.1. Algoritma Genetika

Algoritma genetika merupakan evaluasi atau perkembangan dunia komputer dalam bidang kecerdasan buatan (*artificial intelligence*). Kemunculan algoritma genetika ini terinspirasi oleh teori Darwin (walaupun pada kenyataannya teori tersebut terbukti keliru) dan teori-teori dalam ilmu biologi, sehingga banyak istilah dan konsep biologi yang digunakan dalam algoritma genetika, karena sesuai dengan namanya, proses-proses yang terjadi dalam algoritma genetika sama dengan apa yang terjadi pada evaluasi biologi. [2]

Algoritma genetika adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Algoritma genetika merupakan salah satu algoritma yang sangat tepat digunakan dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan oleh metode konvensional.

2.2.1.1. **Aplikasi Algoritma Genetika**

Sejak pertama kali dirintis oleh John Holland pada tahun 1960-an, algoritma genetika telah dipelajari, diteliti dan diaplikasikan secara luas pada berbagai bidang. Algoritma banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter optimal. [3]

Keuntungan penggunaan algoritma genetika sangat jelas terlihat dari kemudahan implementasi dan kemampuannya untuk menentukan solusi seperti:

1. Ruang masalah sangat besar, kompleks, dan sulit dipahami.
2. Kurang atau bahkan tidak ada pengetahuan yang memandai untuk merepresentasikan masalah ke dalam ruang pencarian yang lebih sempit.
3. Tidak tersedianya analisis matematika yang memadai.
4. Ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi.
5. Solusi yang diharapkan tidak harus paling optimal, tetapi cukup bisa diterima.
6. Terdapat batasan waktu, misalnya dalam real time systems atau system waktu nyata.

Algoritma genetika telah banyak diaplikasikan untuk penyelesaian masalah dan pemodelan dalam bidang teknologi, bisnis dan *entertainment* seperti:[3]

1. Optimasi

   Algoritma Genetika untuk optimasi numeric dan optimasi kombinatorial seperti Traveling Salesman Problem (TSP), perancangan Intergrated Circuit atau IC [LOU93], job shop scheduling [GOL91], optimasi video, dan suara.
2. Pemograman otomatis

   Algoritma genetika telah digunakan untuk melakukan proses evolusi terhadap program komputer untuk merancang struktur komputasional, seperti cellular automatis dan sorting networks.
3. *Machine learning*

Algoritma genetika telah berhasil diaplikasikan untuk memprediksi struktur protein. Algoritma genetika juga berhasil diaplikasikan dalam perancangan *neural networks* (jaringan syaraf tiruan) untuk melakukan proses evolusi terhadap aturan-aturan pada *learning classifier systems* atau *symbolic prosuction systems*. Algoritma genetika juga digunakan untuk mengontrol robot.

4. Model Ekonomi

   Algoritma genetika telah digunakan untuk memodelkan proses-proses inovasi dan pembangunan *bidding strategies*.

5. Model Sistem Imunisasi

   Algoritma genetika telah berhasil digunakan untuk memodelkan berbagai aspek pada sistem imunisasi alamiah, termasuk somatic mulation selama kehidupan individu dan menentukan keluarga dengan gen ganda (*multi - gen families*) sepanjang waktu evolusi.

6. Model Ekologi

   Algoritma genetika telah berhasil digunakan untuk memodelkan fenomena ekologis seperti *host-parasite co-evolutions*, simbiosis dan aliran sumber daya dalam ekologi.

7. Interaksi antara Evolusi dan Belajar

   Algoritma genetika telah digunakan untuk mempelajari bagaimana proses belajar suatu individu bisa mempengaruhi proses evolusi suatu species dan sebaliknya.

2.2.2. Komponen Algoritma Genetika

Ada beberapa komponen algoritma genetika yang perlu diketahui sebelum pembuatan program diantaranya yaitu:[2]

2.2.2.1. **Teknik Pengkodean**

Teknik pengkodean adalah bagaimana mengodekan gen dari kromosom, dimana gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu *variable*, agar dapat diproses melalui algoritma genetik, maka *alternative*

solusi tersebut harus dikodekan terlebih dahulu kedalam bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang mengodekan informasi yang disimpan didalam kromosom.

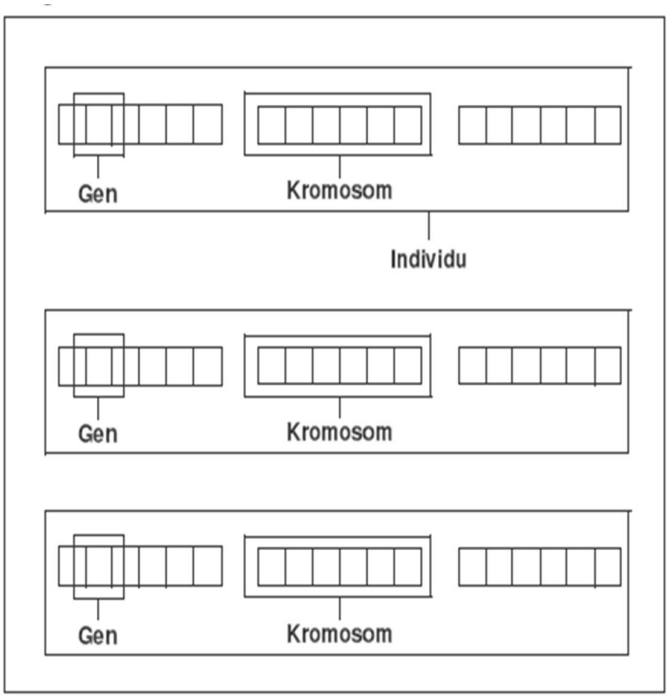
Gen dapat direpresentasikan dalam bentuk : bit, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

2.2.2.2. **Pengertian Individu**

Individu menyatakan salah satu solusi yang mungkin. Individu dapat dikatakan sama dengan kromosom, yang merupakan kumpulan gen. Beberapa definisi penting yang perlu diperhatikan dalam membangun penyelesain masalah menggunakan algoritma genetika, yakni:[2]

1. *Genotype* (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa biner, float, interger maupun karakter, atau kombinatorial
2. Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
3. *Allele*, merupakan nilai dari gen
4. Individu, menyatakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
5. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evalusi.
6. Seleksi, merupakan proses untuk mendapatkan calon individu yang baik.
7. *Crossover*, merupakan proses pertukaran atau kawin silang gen-gen dari dua induk tertentu.

Populasi



Gambar 2.2 Individu dalam algoritma genetik [3]

Hal-hal yang harus dilakukan dalam Algoritma genetika yaitu:

1. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.
2. Mendefinisikan nilai *fitness*, yang merupakan ukuran baik-tidaknya sebuah individu baik-tidaknya solusi yang didapat.
3. Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti *random-walk*.
4. Menentukan proses seleksi yang akan digunakan.
5. Menentukan proses perkawinan silang (*cross-over*) dan mutasi gen yang akan digunakan.

Hal penting yang harus diketahui dalam pemakaian Algoritma genetika:

1. Algoritma genetika adalah algoritma yang dikembangkan dari proses pencarian solusi menggunakan pencarian acak, ini terlihat pada proses pembangkitan populasi awal yang menyatakan sekumpulan solusi yang dipilih secara acak.
2. Berikut ini pencarian dilakukan berdasarkan proses teori genetika yang memperhatikan pemikiran bagaimana memperoleh individu yang lebih baik, sehingga dalam proses evaluasi dapat diharapkan diperoleh individu yang baik.

2.2.2.3. **Membangkitkan Populasi Awal**

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui *procedure* tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal. [3]

Teknik dalam pembangkitan populasi awal ini ada beberapa cara, diantaranya adalah sebagai berikut:

1. Seperti pada metode *random search*, pencarian solusi dimulai dari suatu titik uji tertentu. Titik uji tersebut dianggap sebagai alternatif solusi yang disebut sebagai populasi.
2. *Random Generator*

   Random generator adalah melibatkan pembangkitan bilangan random untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan.
3. Pendekatan tertentu (memasukan nilai tertentu kedalam gen)

   Cara ini adalah dengan memasukan nilai tertentu kedalam gen dari populasi awal yang dibentuk.
4. Permutasi Gen

   Cara ini adalah penggunaan permutasi josephus dalam permasalahan kombinatorial seperti TSP.

2.2.2.4. **Fungsi *fitness***

Suatu individu atau kromosom dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performasinya. Fungsi yang digunakan untuk mengukur nilai kecocokan atau derajat optimalitas suatu kromosom disebut dengan *fitness function.*

Nilai yang dihasilkan dari fungsi tersebut menandakan seberapa optimal solusi yang diperoleh. Nilai yang dihasilkan oleh fungsi *fitness* merepresentasikan seberapa banyak jumlah persyaratan yang dilanggar, sehingga dalam kasus penjadwalan perkuliahan semakin kecil jumlah pelanggaran yang dihasilkan maka solusi yang dihasilkan akan semakin baik.

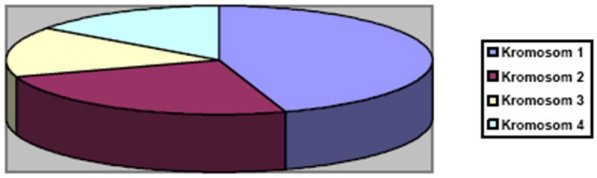
2.2.2.5. **Seleksi**

Setiap kromosom yang terdapat dalam populasi akan melalui proses seleksi untuk dipilih menjadi orang tua. Sesuai dengan teori Evolusi Darwin maka kromosom yang baik akan bertahan dan menghasilkan keturunan yang baru untuk generasi selanjutnya.[3]

Ada beberapa metode seleksi, yaitu:

1. Seleksi *Roulette Wheel*

   Model seleksi ini merupakan model yang paling besar variansinya. Munculnya individu superior sering terjadi pada model ini, sehingga perlu strategi lain menangani hal ini.

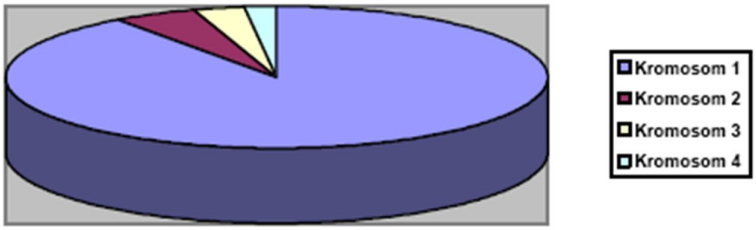


Gambar 2.3 *roulette wheel* [3]

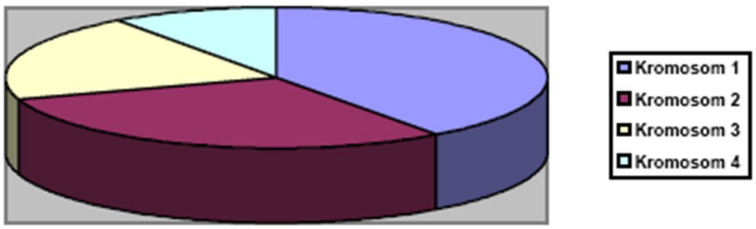
2. Seleksi Rangking

   Seleksi ini memperbaiki proses seleksi yang sebelumnya yaitu *roulette wheel* karena pada seleksi tersebut kemungkinan selain satu kromosom mempunyai nilai *fitness* yang mendominasi hingga 90% biasa terjadi.

Sehingga nilai *fitness* yang lain akan mempunyai kemungkinan yang sangat kecil untuk terpilih. Seleksi rangking dipakai untuk mengatasi masalah di atas, pertama-tama, diurutkan seluruh kromosom berdasarkan bagus-tidaknya solusi berdasarkan nilai *fitness*-nya. Setelah diurutkan, kromosom terburuk diberi nilai *fitness* baru sebesar 1, kromosom kedua terburuk diberi nilai *fitness* baru sebesar 2, dan seterusnya. Kromosom terbaik diberi nilai *fitness* baru sebesar n dimana n adalah banyak kromosom dalam suatu populasi.



Gambar 2.4 sebelum dirangking [3]



Gambar 2.5 sesudah dirangking [3]

3. Seleksi *Steady State*

   Metode ini tidak banyak digunakan dalam proses seleksi karena dilakukan dengan mempertahankan individu yang terbaik. Pada setiap generasi, akan dipilih beberapa kromosom-kromosom yang memiliki nilai *fitness* terburuk akan digantikan dengan *offspring* yang baru. Sehingga pada generasi selanjutnya akan terdapat beberapa populasi yang dipertahankan.

4. Seleksi Turnamen

   Merupakan metode seleksi lainnya yang didasari fenomena alamiah seperti

turnamen antar individu dalam populasi. Dilakukan dengan memilih secara acak beberapa kromosom dari populasi. Individu-individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk.

5. *Truncation Random*

   Metode ini lebih mudah diterapkan jika dibandingkan dengan metode *Roulette Wheel,* pemilihan kromosom dilakukan secara acak tetapi tidak semua kromosom mendapatkan kesempatan tersebut, hanya kromosom terbaik saja yang berpeluang.

2.2.2.6. ***Cross-Over* atau Kawin Silang**

Proses kawin silang adalah salah satu operator penting dalam algoritma genetika, metode dan tipe *crossover* yang dilakukan tergantung dari *encoding* dan permasalahan yang diangkat. Ada beberapa cara yang bisa digunakan untuk melakukan *crossover* sesuai dengan encodingnya sebagai berikut:[2]

1. Binary encoding
   a. *Crossover* satu titik

      Memilih satu titik tertentu, selanjutnya nilai biner sampai titik *crossover*nya dari induk pertama digunakan dan sisanya dilanjutkan dengan nilai biner dari induk kedua.

      Contoh

      11001011 + 1101**1111** = **11001111**

   b. *Crossover* dua titik

      Memilih dua titik tertentu, lalu nilai biner sampai titik *crossover* pertama pada induk pertama digunakan, dilanjutkan dengan nilai biner dari titik sampai titik kedua dari induk kedua, kemudian sisanya dilanjutkan nilai biner dari titik kedua induk pertama lagi.

      Contoh:

      11001**011** + 11**011**11 = **11011111**

   c. *Crossover uniform*

      Nilai biner yang digunakan dipilih secara random dari kedua induk.

      Contoh

1**10**01011 + 11011**11** = **11011111**

d. *Crossover* aritmatika

Suatu operasi aritmatika digunakan untuk menghasilkan *offspring* yang baru.

Contoh:

11001011 + 11011111 = 11001001 (**AND**)

2. *Permutation encoding*

Memilih satu titik tertentu, nilai *permutation* sampai titik *crossover*. Pada induk pertama digunakan lalu sisanya dilakukan *scan* terlebih dahulu, jika nilai permutasi pada induk kedua belum ada pada *offspring* nilai tersebut ditambahkan.

Contoh:

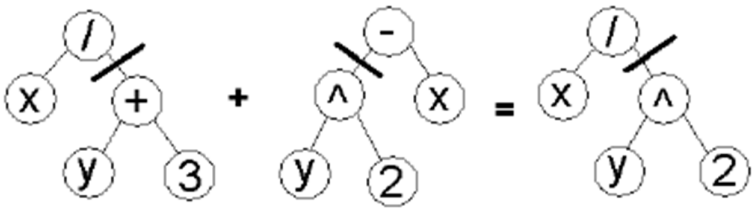
(**12345**6789) + (4**53689**721) = **12345689**

3. *Value encoding*

Semua metode *crossover* pada binary *crossover* bisa digunakan.

4. *Tree encoding*

Memilih satu titik tertentu dari tiap induk, dan menggunakan *tree* dibawah titik pada induk pertama dan *tree* dibawah induk kedua.



Gambar 2.6 *Tree encoding*.[3]

2.2.2.7. **Mutasi**

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Mutasi ini berperan untuk menggantikan gen yang hilang dari populasi akibat seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Beberapa cara operasi mutasi yang diterapkan dalam algoritma genetika, antara lain: [3]

1. Mutasi dalam pengkodean biner.

   Mutasi pada pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang dipilih secara acak pada kromosom.

   Contoh.

   Kromosom sebelum mutasi : 1 0 0 1 0 **1** 1 1

   Kromosom sesudah mutasi : 1 0 0 1 0 **0** 1 1

2. Mutasi dalam pengkodean permutasi.

   Proses mutasi yang dilakukan pengkodean biner dengan mengubah langsung bit pada kromosom tidak dapat dilakukan pada pengkodean permutasi karena konsistensi urutan harus diperhatikan, salah satu cara yang dapat dilakukan adalah dengan memilih dua posisi dari kromosom dan kemudian nilainya saling tukar.

   Contoh:

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

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

3. Mutasi dalam pengkodean nilai

   Proses mutasi dalam pengkodean nilai dapat dilakukan dengan berbagai cara, salah satunya yaitu dengan memilih sembarang posisi gen pada kromosom, nilai yang ada tersebut kemudian ditambah atau dikurangi dengan suatu nilai terkecil tertentu yang diambil secara acak.

   Contoh mutasi dalam pengkodean nilai riil dengan yang ditambahkan atau dikurangkan adalah 0,1

Contoh:

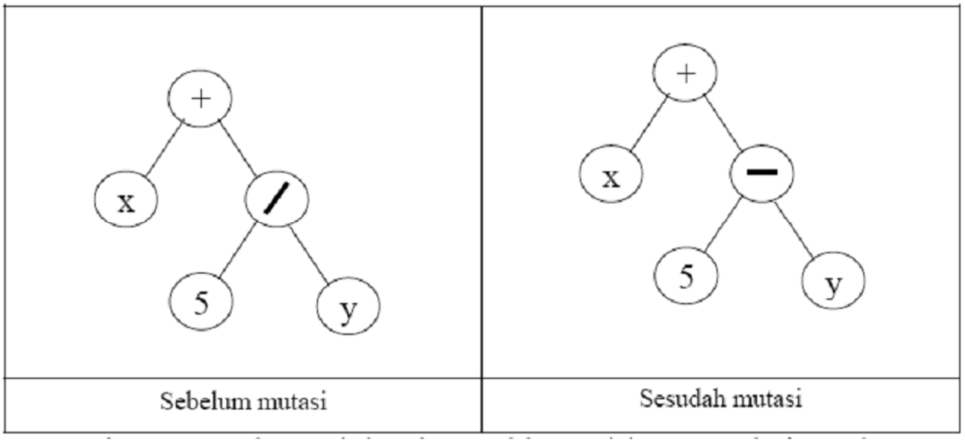
Kromosom sebelum mutasi : 1,43 1,09 4,51 9,11 6,94

Kromosom sesudah mutasi : 1,43 1,19 4,51 9,01 6,94

4. Mutasi dalam pengkodean pohon

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

   Contoh



Gambar 2.7 Mutasi dalam pengkodean pohon [3]

2.2.2.8. **Kondisi Selesai**

Jika kondisi telah terpenuhi, maka algoritma genetika akan menghentikan proses pencariannya, tetapi jika belum terpenuhi maka algoritma genetika akan kembali ke evaluasi *fitness*.

2.2.3. Android

Android merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, middleware dan aplikasi inti yang di-release oleh Google. Sedangkan Android SDK (Software Development Kit) menyediakan Tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman Java.[1]

Dikembangkan bersama antara Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, NVIDIA yang tergabung dalam OHA (*Open Handset Alliance*) dengan tujuan membuat sebuah standar terbuka untuk perangkat bergerak (mobile device).

2.2.4. Internet

Pengertian Internet, internet (*interconnected computer networks*) bisa didefinisikan *network* komputer tiada batas yang menjadi penghubung pengguna komputer dengan pengguna komputer lainnya serta dapat berhubungan dengan komputer di sebuah wilayah ke wilayah di penjuru dunia, dimana di dalam jaringan tersebut mempunyai berbagai macam informasi serta fasilitas layanan internet *browsing* atau *surfing*. Istilah ini lebih dikenal dengan "*online*" di internet. Pekerjaan ini bisa diibaratkan seperti kita berjalan-jalan di tempat hiburan sembari melihat-lihat ke toko-toko namun tidak membeli jualan tersebut.

2.2.5. *Tools* Implementasi

Tools implementasi merupakan peralatan-peralatan yang dibutuhkan dalam membangun perangkat lunak. Adapun *tools* implementasi yang digunakan untuk membuat aplikasi optimasi penjadwalan berbasis Android adalah JDK sebagai perangkat pembangun Java, Eclipse IDE sebagai lingkungan pengembangan perangkat lunak, MySQL sebagai sistem manajemen basis data XAMPP sebagai server pada localhost.

2.2.5.1. **JDK**

JDK (Java Development Kit) adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemrograman Java. JDK berjalan diatas sebuah virtual machine yang dinamakan JVM (Java Virtual Machine)**.** [5]

Dokumentasi JDK berisi spesifikasi API, deskripsi fitur, panduan pengembang, referensi halaman untuk perkakas JDK dan utilitas, demo, dan link ke informasi terkait. Dokumentasi ini juga tersedia dalam bundel yand dapat diunduh dan diinstal pada komputer.

2.2.5.2. **Eclipse IDE**

Eclipse adalah sebuat komunitas proyek *open source* yang difokuskan pada membangun sebuah *platform* pengembangan yang dapat diperluas, *runtime* dan kerangka aplikasi untuk membangun, menyebarkan dan mengelola perangkat lunak di seluruh daur hidup perangkat lunak.

Komunitas Eclipse memiliki lebih dari 60 proyek *open source*. Proyek-proyek ini secara konsep terbagi menjadi 7 kategori : [4]

1. *Enterprise Development*
2. *Embedded and Device Development*
3. *Rich Client Platform*
4. *Rich Internet Applications*
5. *Application Frameworks*
6. *Application Lifecycle Management* (ALM)
7. *Service Oriented Architecture* (SOA)

Komunitas Eclipse juga didukung oleh ekosistem besar dan dinamis dari penyedia solusi teknologi informasi utama, *start-up* inovatif, universitas dan lembaga penelitian, dan individu yang memperluas, mendukung, dan melengkapi *platform* Eclipse.

2.2.5.3. **MySQL**

MySQL adalah sebuah perangkat lunak iasm manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta iasm atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia

yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.[10]

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu iasm *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL iasm sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

2.2.5.4. **XAMPP**

XAMPP merupakan singakatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, Perl. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FTP *Server*, phpMyAdmin dan berbagai pustaka bantu lainnya. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server* Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkofigurasikannya secara otomatis.

2.2.6. UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah himpunan struktur dan teknik untuk pemodelan desain program

berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.2.6.1. **Diagram UML**

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:[9]

1. Use Case Diagram untuk memodelkan proses bisnis.
2. Conceptual Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. Sequence Diagram untuk memodelkan pengiriman pesan (message) antar objects.
4. Collaboration Diagram untuk memodelkan interaksi antar objects.
5. State Diagram untuk memodelkan perilaku objects di dalam sistem.
6. Activity Diagram untuk memodelkan perilaku Use Cases dan objects di dalam system.
7. Class Diagram untuk memodelkan struktur kelas.
8. Object Diagram untuk memodelkan struktur object.
9. Component Diagram untuk memodelkan komponen object.

10. Deployment Diagram untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi object, yaitu use case diagram, sequence diagram, collaboration diagram, dan class diagram.

2. *Use Case* Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. Aktor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh aktor. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. Aktor yang melakukan operasi dihubungkan dengan garis lurus ke *usecase*.

3. Sequence Diagram

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

4. Collaboration Diagram

Collaboration diagram dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence diagram* yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration* diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

5. Class Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

2.2.7. JAVA

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai computer termasuk telepon genggam. Dikembangkan oleh Sun Microsystems dan diterbitkan tahun 1995. Java tidak boleh disalahpahami sebagai JavaScript. JavaScript adalah bahasa *scripting* yang digunakan oleh web *browser*.[7]

2.2.7.1. **Versi Awal**

Versi awal Java ditahun 1996 sudah merupakan versi release sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

1. Java.lang: Peruntukan kelas elemen-elemen dasar.
2. Java.io: Peruntukan kelas input dan output, termasuk penggunaan berkas.
3. Java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
4. Java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. Java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
6. Java.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

2.2.7.2. **Kelebihan**

Beberapa kelebihan menggunakan bahasa pemrograman Java adalah sebagai berikut:

1. Multiplatform. Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating

system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft 35 Windows, Linux, Mac OS dan Sun Solaris. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk menginterpretasikan bytecode tersebut.

2. OOP (Object Oriented Programming - Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di Java adalah Objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut Object. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan Java sebagai salah satu bahasa pemograman termudah, bahkan untuk fungsi fungsi yang advance seperti komunikasi antara komputer sekalipun.
3. Perpustakaan Kelas yang Lengkap, Java terkenal dengan kelengkapan library/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
4. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.2.7.3. **Kekurangan**

Beberapa kekurangan menggunakan bahasa pemrograman Java adalah sebagai berikut:

1. Tulis sekali, perbaiki di mana saja - Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.
2. Mudah didekompilasi. proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-reverse-engineer.
3. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.2.7.4. **Tahap Kompilasi Java**

Poin-poin dalam tahapan kompilasi Java adalah sebagai berikut:

1. Tulis / Ubah. Pemrogram menulis program dan menyimpannya di media dalam bentuk berkas *'.java'*.
2. Kompilasi. Pengkompilasi membentuk *bytecodes* dari program menjadi bentuk berkas *'.class'*.
3. Muat. Pemuat kelas memuat *bytecodes* ke memori.
4. Verifikasi. Peng-verifikasi memastikan *bytecodes* tidak mengganggu sistem keamanan Java.
5. Jalankan. Penerjemah menerjemahkan *bytecodes* ke bahasa mesin tidak bisa dipakai.