BAB II

LANDASAN TEORI

2.1 Jadwal

Jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau rencana kegiatan dengan pembagian waktu pelaksanaan terperinci, sedangkan penjadwalan adalah proses, cara, pembuatan menjadwalkan atau memasukkan dalam jadwal (Lee, 2000). Berdasarkan Kamus Besar Bahasa Indonesia, jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja. Jadwal juga didefinisikan sebagai daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan penjadwalan merupakan prose penyusunan daftar pekerjaan yang akan dilakukan untuk mencapai atau mewujudkan suatu tujuan tertentu yang juga memuat tabel waktu pelaksanaan.

2.2 Penjadwalan Mata Kuliah

Masalah penjadwalan dapat diklasifikasikan ke dalam dua kategori utama, yaitu masalah penjadwalan kuliah dan masalah penjadwalan ujian (Gunawan, Ong dan Ng, 2004). Penjadwalan mata kuliah merupakan proses penentuan jadwal mata kuliah dan berapa jadwal yang diselenggarakan pada semester yang bersangkutan. Biasanya penjadwalan kuliah ini dilaksanakan sebelum perwalian berlangsung. Tujuan utama dari penjadwalan kuliah adalah untuk menentukan dan mengatur jadwal mata kuliah yang akan diselenggarakan pada semester yang akan diselanggarakan selanjutnya. Penjadwalan mata kuliah

salah satu kegiatan yang sangat penting untuk dapat terlaksananya sebuah proses belajar mengajar yang baik di perguruan tinggi.

2.3 Aplikasi

Definisi aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi suatu pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komnputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu (Noviansyah, 2008). Jadi aplikasi bisa diartikan sebagai program komputer yang ditulis dalam suatu bahasa pemrograman dan dipergunakan untuk menyelesaikan masalah tertentu. Aplikasi *software* yang dirancang untuk suatu tugas khusus dapat dibedakan menjadi dua jenis, yaitu :

1. Aplikasi *softawre* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi *software* paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.4 Aplikasi Web

Menurut Janner Simamarta (2010), aplikasi web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis *web.* Jadi aplikasi web adalah suatu aplikasi yang diakses dan berjalan di atas *platform browser* dengan melalui internet atau intranet. Aplikasi web juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang di dukung oleh *browser* yaitu seperti HTML, JavaScript, AJAX, Java, dan lain-lain. Aplikasi web memiliki beberapa kemudahan dan keuntungan antara lain yaitu:

a. Bisa diakses dari mana saja. Karena aplikasi terpasang di server, maka aplikasi tersebut bisa diakses dari mana saja dan dengan komputer apa saja.
b. Bisa digunakan pada sistem operasi apa pun *(Multi platform).* Karena berbasis internet/intranet dan diakses melalui *browser*, maka aplikasi tersebut bisa diakses dengan sistem operasi apa pun.
c. Aplikasi yang diperlukan hanyalah *browser* (Mozilla Firefox, Internet Explorer maupun *browser* lainnya), sehingga tidak perlu memasang program lain untuk menggunakan aplikasi web ini.
d. Selalu mendapatkan versi terbaru dari aplikasi. Karena aplikasi tersebut terpasang di *server* intranet/internet, perusahaan pembuat aplikasi bisa memperbarui aplikasinya terus-menerus. Saat mengakses aplikasi pasti yang didapatkan adalah aplikasi versi terbaru. Tidak perlu lagi melakukan *upgrade*.

2.5 Algoritma Genetika

Algoritma Genetika merupakan metode *heuristic adaptif* yang memiliki dasar pemikiran atau gagasan untuk proses seleksi alam dan genetika berdasarkan penelitian Charles Darwin. Dengan kata lain pencarian solusi suatu masalah dengan Algoritma Genetika akan terus berevolusi (Kusumadewi dan Purnomo, 2005). Algoritma ini didasarkan pada proses genetika yang ada dalam makhluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam (siapa yang kuat, dia yang bertahan). Dengan meniru teori evolusi ini, Algoritma Genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan yang ada dalam dunia nyata.

Algoritma Genetika pertama kali ditemukan dan mulai dirintis pada tahun 1960 oleh John Holland dari University of Michigan. Setiap masalah yang

berbentuk adaptasi (alam maupun buatan) dapat diformulasikan dalam teknologi genetika. Dari kemudahan implementasi dan kemampuannya untuk menentukan solusi, Algoritma Genetika dapat menentukan solusi permasalahan-permasalahan dengan karakteristik seperti di bawah ini (Suyanto,2005):

a. Ruang masalah sangat besar, kompleks, dan sulit dipahami.
b. Kurang atau bahkan tidak ada pengetahuan yang menandai untuk mempresentasikan masalah ke dalam ruang pencarian yang lebih sempit.
c. Tidak tersedianya analisis matematika yang memadai.
d. Ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi.
e. Solusi yang diharapkan tidak harus paling optimal, tetapi cukup bisa diterima.
f. Terdapat batasan waktu.

Beberapa definisi penting yang perlu diperhatikan dalam mendefinisikan individu untk membangun penyelesaian permasalahan dengan algoritma adalah sebagai berikut :

a. Genotype (Gen), merupakan 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 nilai biner, float, integer maupun karakter, atau kombinatorial.
b. Allele, merupakan nilai yang berada dalam gen.
c. Kromosom, merupakan gabungan dari gen-gen yang membentuk nilai tertentu.
d. Individu, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.

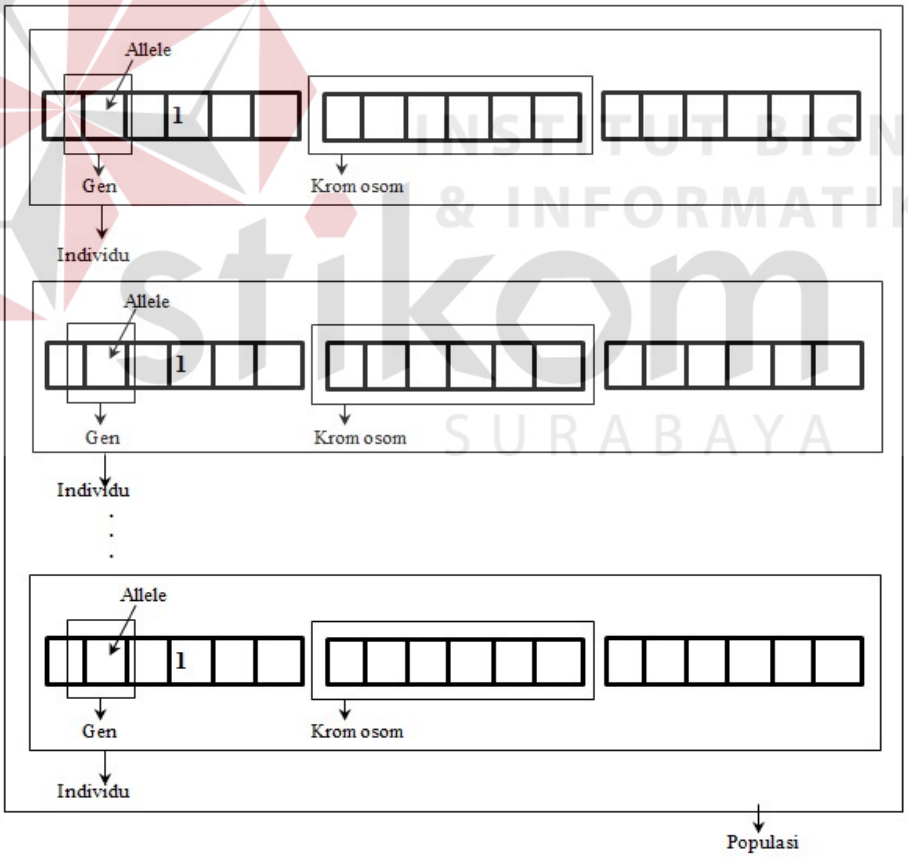
e. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.

f. Generasi, menyatakan satu siklus proses evolusi atau satu iterasi di dalam Algoritma Genetika.

g. Seleksi, merupakan proses untuk mendapatkan calon individu yang baik.

h. *Crossover,* merupakan proses pertukaran atau kawin silang gen-gen dari induk tertentu.

Untuk lebih jelasnya diilustrasikan perbedaan istilah-istilah di atas pada gambar 2.1.



Gambar 2.1 Ilustrasi representasi penyelesaian permasalahan dalam Algoritma Genetika

Sebelum memanfaatkan Algoritma Genetika, hal yang harus dilakukan adalah menyandikan solusi dari masalah yang diberikan ke dalam kromosom pada Algoritma Genetika dan membandingkan nilai *fitness*-nya. Sebuah representasi Algoritma Genetika yang efektif dan nilai *fitness* yang bermakna adalah kunci keberhasilan dalam aplikasi genetika. Algoritma Genetika bekerja dengan menggunakan pendekatan random, dan nilai-nilai yang dihasilkan adalah nilai random. Pada kasus penjadwalan ini dengan model genetika ditujukan untuk mendapatkan kombinasi yang tepat antara variabel dosen, waktu, dan ruang.

Semakin banyak iterasi yang dilakukan, maka waktu yang dibutuhkan akan semakin lama. Beberapa hal yang harus dilakukan dalam Algoritma Genetika adalah sebagai berikut:

a. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.
b. Mendefinisikan nilai *fitness*, yang merupakan ukuran baik atau tidaknya sebuah individu.
c. Menentukan proses pembangkitan populasi awal, hal ini biasanya dilakukan dengan menggunakan pembangkitan secara acak.
d. Menentukan proses seleksi yang akan digunakan.
e. Menentukan proses perkawinan silang dan mutasi gen yang akan digunakan.

Untuk lebih jelasnya dapat dilihat pada gambar *flowchart* algoritma genetika secara umum yaitu pada gambar 2.2.

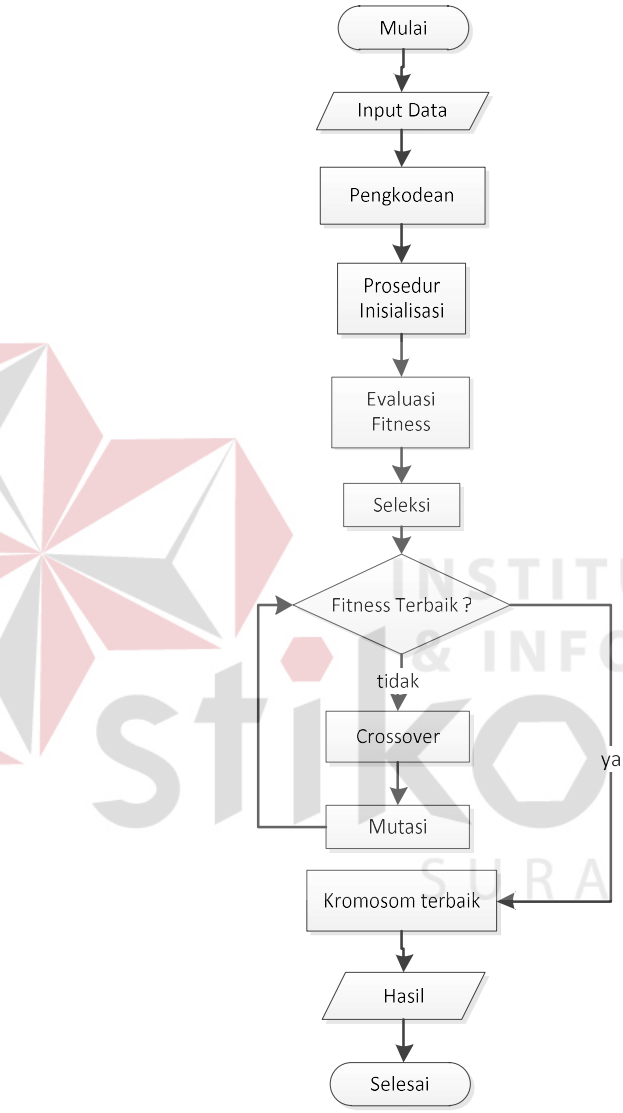
Komponen-komponen utama Algoritma Genetika yaitu:

1. Teknik Pengodean
2. Membangkitkan Populasi Awal

3. Seleksi

4. Pindah Silang *(Crossover)*

5. Mutasi



Gambar 2.2 *Flowchart* Algoritma Genetika secara umum

2.5.1 Teknik Pengodean

Teknik pengodean adalah bagaimana mengodekan gen dari kromosom, dimana gen merupakan bagian dari kromosom dan satu gen biasanya akan

mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk bit, bilangan real, daftar aturan, elemen program dan represesntasi lainnya yang dapat diimplementasikan untuk operator genetika. Dengan demikian kromosom dapat direpresentaikan dengan menggunakan :

a. String bit : 100110 dst.

b. Array bilangan real : 65.65,-67.99,76.44 dst.

c. Elemen permutasi : E2, E6, E5 dst.

d. Elemen program : pemrograman genetika

e. Struktur lainnya

2.5.2 Membangkitkan Populasi awal

Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi (Hendarto, 2002). Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak melalui prosedur 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.

2.5.3 Seleksi

Seleksi digunakan untuk memilih individu-individu mana saja yang akan dipilih untuk proses kawin silang dan mutasi. Seleksi juga digunakan untuk mendapatkan calon induk yang baik. Induk yang baik akan menghasilkan keturunan yang baik. Semakin tinggi nilai *fitness* suatu individu semakin besar kemungkinan untuk terpilih. Nilai *fitness* diperoleh dari fungsi *fitness* yang

ditentukan oleh nilai *penalty*. *Penalty* tersebut menunjukkan jumlah pelanggaran kendala pada suatu kromosom. Semakin tinggi nilai *fitness* akan semakin besar kemungkinan kromosom tersebut terpilih ke generasi berikutnya. Jadi nilai *penalty* berbanding terbalik dengan nilai *fitness,* semakin kecil nilai *penalty* (jumlah pelanggaran) semakin besar nilai *fitness* nya. Berikut fungsi *fitness* secara umum.

$$Fitness = \frac{1}{1+penalty} \qquad (2.1)$$

Nilai *fitness* ini yang nantinnya akan digunakan pada tahap-tahap seleksi berikutnya. Terdapat beberapa metode seleksi, disini hanya dibahas dua metode yaitu mesin *roullete*, dan turnamen.

a. Seleksi dengan mesin *roulette*

Metode seleksi mesin roulette ini merupakan metode yang paling sederhana dan sering juga dikenal dengan nama *stochastic sampling with replacement*. Cara kerja metode ini adalah sebagai berikut :

1. Dihitung nilai *fitness* dari masing-masing individu (*fi*, dimana I adalah individu ke-1 s/d ke-n)
2. Dihitung total *fitness* semua individu
3. Dihitung probabilitas masing-masing individu
4. Dari probabilitas tersebut, dihitung jatah masing-masing individu pada angka 1 sampai 100
5. Dibangkitkan bilangan random antara 1-100
6. Dari bilangan random yang dihasilkan, ditentukan individu mana yang terpilih dalam proses seleksi

b. Seleksi dengan turnamen

Pada metode seleksi turnamen, ditetapkan suatu nilai tour untuk individu-individu yang dipilih secara random dari suatu populasi. Individu-individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk. Parameter yang digunakan pada metode ini adalah ukuran yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi).

2.5.4 Pindah Silang (*Crossover*)

Pindah silang (*crossover*) adalah operator dari Algoritma Genetika yang melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan titik baru dalam ruang pencarian yang siap untuk diuji. Operasi ini tidak selalu dilakuan pada semua individu yang ada. Individu dipilih secara acak untuk dilakukan crossing. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan. Prinsip dari pindah silang ini adalah melakukan operasi (pertukaran, aritmatika) pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan individu baru. Proses *crossover* dilakukan pada setiap individu dengan probabilitas *crossover* yang ditentukan.

a. *One point crossover*

Sebuah titik *crossover* dipilih, selanjutnya string mulai dari awal kromosom sampai dengan titik tersebut disalin dari salah satu orang tua ke keturunannya, kemudian sisa keturunan disalin dari orangtua yang kedua.

b. *Two point crossover*

Dua titik *crossover* dipilih, selanjutnya string mulai dari awal kromosom sampai dengan titik *crossover* pertama disalin dari salah satu orangtua ke

keturunannya kemudian mulai dari titik *crossover* pertama sampai dengan titik kedua disalin dari orangtua kedua. Sisanya disalin dari orangtua pertama.

2.5.5 Mutasi

Pada Algoritma Genetika, operator berikutnya adalah mutasi gen. Operator ini berperan unutk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Kromosom anak dimutasi dengan menambahkan nilai random yang sangat kecil (ukuran langkah mutasi), dengan probabilitas yang rendah. Probabilitas mutasi (p_m) didefinisikan sebagai presentasi dari jumlah total gen pada populasi yang mengalami mutasi. Jika probabilitas mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila probabilitas mutasi ini terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dengan induknya. Kromosom hasil mutasi harus diperiksa, apakah masih berada pada domain solusi, dan bila perlu bisa dilakukan perbaikan.

Cara sederhana untuk mendapatkan mutasi adalah dengan mengganti satu atau beberapa nilai gen dari kromosom. Langkah-langkahnya adalah sebagai berikut :

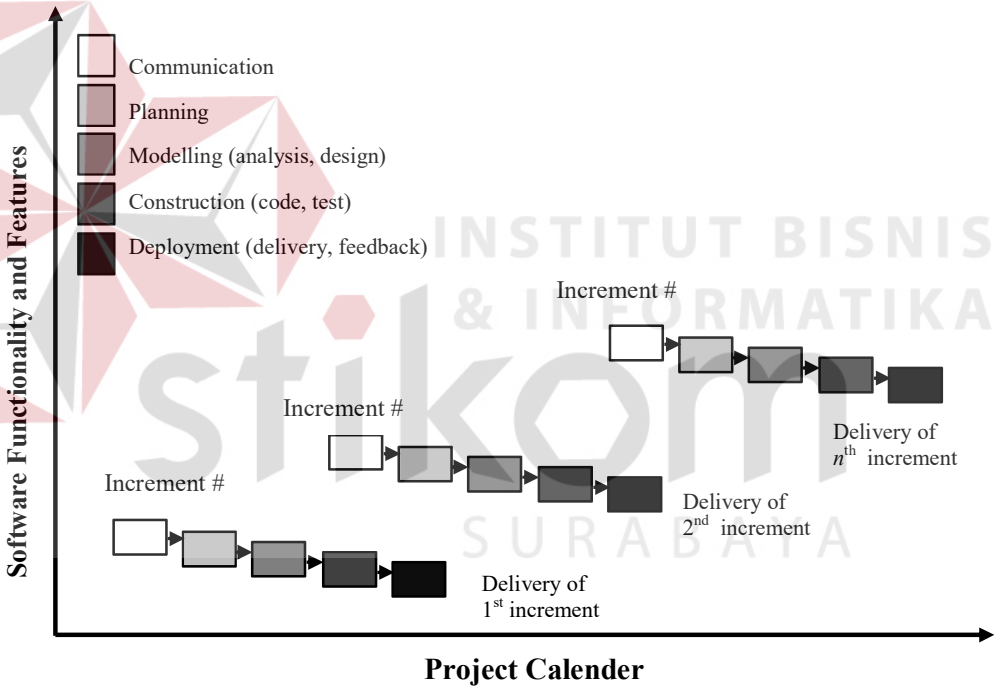
1. Hitung jumlah gen pada populasi (panjang kromosom dikalikan dengan ukuran populasi)
2. Pilih secara acak gen yang akan dimutasi
3. Tentukan kromosom dari gen terpilih untuk dimutasi
4. Ganti nilai gen(0 ke 1,atau 1 ke 0) dari kromosom yang akan dimutasi tersebut

Contoh : 1**0**01101 → 1**1**01101

2.6 *Software Development Life Cycle* (SDLC)

Software Development Life Cycle (SDLC) merupakan sebuah rangkaian proses hidup dari sebuah perangkat lunak, mulai dari analisis hingga sebuah perangkat lunak tidak terpakai lagi. Perangkat lunak tersebut dinyatakan hidup kembali dalam sebuah revisi atau pengembangan baru. Bentuk SDLC yang digunakan adalah model *incremental*.

2.6.1 Incremental



Gambar 2.3 Model *Incremental* (Pressman, 2015).

Model *incremental* (model penambahan sedikit demi sedikit) merupakan suatu model proses yang dirancang untuk menghasilkan perangkat lunak dengan teknik sedikit demi sedikit. Gambar 2.3 menunjukkan tahapan umum dari model *Incremental*. Model ini disebut dengan *incremental* karena hasil pertama

seringkali berupa produk inti *(core product)*, yaitu bahwa spesifikasi kebutuhan dasar perangkat lunak telah ada, tetapi fitur-fitur tambahan tetap belum terselesaikan (Pressman, 2015).

Model *incremental* melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem perangkat lunak yaitu tahap *communication*, *planning*, *modeling*, *construction* dan *deployment* Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam model *incremental* (Pressman, 2015) :

1. *Communication* (komunikasi)

   Langkah ini merupakan analisis terhadap kebutuhan *software*, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *customer*, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

2. *Planning* (perencanaan)

   Proses *planning* merupakan lanjutan dari proses *communication (analysis requirement)*. Tahapan ini menggambarkan tugas-tugas teknis yang dilakukan, sumber daya yang dibutuhkan, produk yang harus dihasilkan, dan jadwal-jadwal kerja termasuk rencana yang akan dilakukan.

3. *Modeling* (pemodelan)

   Proses *modeling* ini akan menerjemahkan syarat kebutuhan-kebutuhan menjadi sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktural data, arsitektur *software*, representasi *interface,* dan detail (algoritma) prosedural.

4. *Construction* (konstruksi)

Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap perangkat lunak yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap perangkat lunak tersebut untuk kemudian bisa diperbaiki.

5. *Deplyoment* (pengoperasian)

Tahapan ini bisa dikatakan akhir dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem perangkat lunak yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

2.6.2 Analisis Sistem

Definisi analisis sistem menurut Fatta (2007), adalah bagaimana memahami dan menspesifikasi dengan detail apa yang harus dilakukan oleh sistem. Menurut Hartono (2005), analisis sistem didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Tahap ini merupakan tahap yang kritis dan sangat penting, karena

kesalahan dalam tahap ini menyebabkan kesalahan pada tahap selanjutnya. Tugas utama dari menganalisis sistem meliputi:

1. Memberikan pelayanan kebutuhan informasi kepada fungsi manajerial di dalam pengendalian pelaksanaan kegiatan operasional perusahaan.
2. Membantu para pemngambil keputusan.
3. Mengevaluasi sistem yang telah ada.
4. Merumuskan tujuan yang ingin dicapai berupa pengolahan data maupun pembuatan laporan baru.
5. Menyusun suatu tahap rencana pengembangan sistem.

2.6.3 Perancangan Sistem

Definisi perancangan sistem menurut Fatta (2007), adalah spesifikasi umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis. Menurut Hartono (2005), desain sistem adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah dari suatu kesatuan yang utuh dan berfungsi. Berdasarkan dua definisi perancangan di atas maka dapat disimpulkan bahwa perancangan adalah penggambaran, perencanaan dan pembuatan sketsa yang kemudian diterjemahkan ke dalam sebuah konsep rancangan sebagai pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis.

2.6.4 Pemodelan Proses

Menurut Fatta (2007), pemodelan proses menggambarkan aktivitas yang dilakukan dan data dapat berpindah di antara aktivitas-aktivitas itu. Cara yang populer untuk merepresentasikan proses model adalah dengan menggunakan *data*

flow diagram (DFD). Ada empat elemen yang menyusun suatu DFD, yaitu proses, *data flow*, *data store* dan *external entity*.

2.6.5 Pemodelan Data

Menurut Fatta (2007), model data adalah cara formal untuk menggambarkan data yang digunakan dan diciptakan dalam suatu sistem bisnis. Model data terbagi atas model data logika dan model data fisik. Model data logika menunjukkan pengaturan data tanpa mengindikasikan data tersebut disimpan, dibuat dan dimanipulasi, sedangkan data fisik menunjukkan data sebenarnya disimpan dalam *database* atau *file.* Salah satu cara pemodelan data adalah dengan *entity relationship diagram* (ERD). ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan dan digunakan dalam sistem. ERD digunakan untuk menunjukkan aturan-aturan bisnis yang ada pada sistem informasi.

2.6.6 *Database*

Menurut Raharjo (2011), *database* adalah sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat. Selain berisi data, *database* juga berisi *metadata*. *Metadata* adalah data yang menjelaskan tentang struktur dari data itu sendiri. *Database* memiliki beberapa model, diantaranya adalah model relasional. Dalam model relasional, tabel-tabel yang terdapat dalam suatu *database* idealnya harus saling berelasi. Tabel merupakan suatu entitas yang tersusun atas kolom dan baris. Dalam dunia *database*, kolom disebut *field* dan baris disebut *record*.

2.7 Testing dan Implemetasi

Testing dan implementasi yaitu tahap mendemonstrasikan dan menjalankan sistem perangkat lunak yang telah selesai dibuat, apakah sudah sesuai dengan kebutuhan yang telah dispesifikasikan. Tahapan ini tertuang dalam suatu dokumen *Test Plan,* yang dimulai dari membuat *Software Testing fundamentals* yang berisi tentang penjelasan penting mengenai *terminology testing.* Selanjutnya merancang *Test Levels* yang terbagi antara target pengetesan dan objektif dari pengetesan. Pada tahap berikutnya adalah mendefinisikan *Test Techniques,* yaitu tentang bagaimana teknik yang digunakan termasuk dasar-dasar pengetesan berdasarkan intuisi dan pengalaman serta teknik pengetesan secara teknik *coding,* teknik kesalahan, teknik penggunaan, dan teknik terkait lainnya. Tahap selanjutnya adalah mendefinisikan *Test – Related Measures,* yaitu ukuran-ukuran pencapaian testing yang telah dilakukan untuk kemudian dievaluasi kembali. Tahap terakhir adalah mendefinisikan *test Process* yang berisi tentang aktivitas testing. (Bertolino and Marchetti, 2004).

Menurut Fatta (2007), pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan *unit testing*, yaitu:

1. *Black Box Testing*

   Terfokus pada unit program yang memenuhi kebutuhan (*requirement*) yang disebutkan dalam spessifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan.

2. *White Box Testing*

White box testing adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis terdapat kesalahan atau tidak.