BAB II

TINJAUAN PUSTAKA

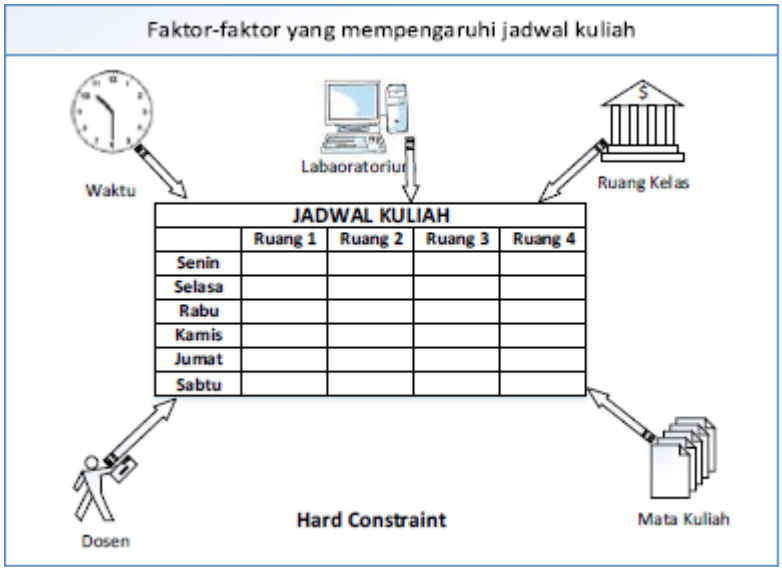
2.1 Kajian Pustaka

Banyak algoritma atau metode yang dapat digunakan untuk mencari solusi-solusi yang mungkin digunakan dalam kasus penjadwalan ini. (Ivan Nugraha , 2008) meneliti algoritma genetika pada penjadwalan kampus yang berjudul "Aplikasi Algoritma genetika Untuk Optimalisasi Penjadwalan Kegiatan Belajar Mengajar". Hasil penelitian menunjukkan beberapa kelemahan yakni terdapat lima matakuliah yang berdekatan dan algoritma genetika bekerja dengan bilangan acak pada kromosom awal, sehingga memungkinkan kromosom terbaik tidak terlibat dalam proses. (Nia Kurnia, 2006) meneliti algoritma genetika pada penjadwalan ujian yang berjudul "Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika". Hasil penelitian menunjukkan rata-rata nilai *fitness* akan semakin tinggi dengan semakin besar probabilitas perkawinan silang, namun pada kondisi tertentu nilai *fitness* akan kembali rendah. (Vinny, dkk, 2013) meneliti algoritma penjadwalan perkuliahan yang berjudul "Optimasi Penjadwalan Perkuliahan dengan Menggunakan Algoritma Genetika". Hasil penelitian menunjukkan suatu jadwal perkuliahan yang optimal. Suatu jadwal dapat dikatakan optimal apabila tidak terdapat pinalti dan mencapai *fitness* terbaik.

2.2 Penjadwalan Matakuliah

Penjadwalan merupakan proses untuk menyusun suatu jadwal atau urutan proses yang diperlukan dalam sebuah permasalahan. Permasalahan penjadwalan biasanya berhubungan dengan penjadwalan kelas dalam sekolah, perkuliahan dan juga dalam lingkup yang tidak jauh berbeda seperti penjadwalan pelajaran sekolah, penjadwalan ujian, atau dapat juga penjadwalan karyawan, baik dalam suatu perusahaan ataupun dalam rumah sakit. Dalam penjadwalan kuliah, yang dibahas adalah tentang pembagian jadwal untuk tiap mahasiswa pada kuliah tertentu sekaligus dosen pengajarnya.

Penjadwalan matakuliah *(lecture timetabling)* di universitas merupakan suatu pekerjaan yang tidak mudah. Permasalahan tersebut sering disebut dengan *University Timetabling Problems* (UTP). Penjadwalan matakuliah merupakan proses pengaturan jadwal dengan memperhatikan dosen, ruang kelas, mata kuliah dan waktu yang disesuaikan dengan sejumlah batasan tertentu. Pada penjadwalan mata kuliah sejumlah matakuliah harus dijadwalkan ke dalam ruang dan slot (pembagian) waktu tertentu dimana penjadwalan tersebut memperhatikan aturan-aturan dan batasan penjadwalan yang telah ditentukan. Berikut ini diberikan gambar komponen yang mempengaruhi penjadwalan mata kuliah :



Gambar 2.1 Komponen yang Mempengaruhi Penjadwalan Matakuliah

Berdasarkan Gambar 2.1, penjadwalan matakuliah dipengaruhi oleh beberapa komponen yang terdiri atas: dosen, ruang kelas, mata kuliah, laboratorium dan waktu dengan sejumlah batasan-batasan tertentu, dimana batasan-batasan tersebut ada yang harus dipenuhi atau tidak boleh dilanggar. Batasan tersebut merupakan ukuran kualitas dari penjadwalan mata kuliah, sehingga suatu jadwal mata kuliah yang optimal dapat terbentuk (Hanita, 2011).

Dalam proses penyelesaian masalah penjadwalan mata kuliah terdapat *constraint* yang tidak boleh dilanggar. *Constraint* tersebut merupakan ukuran kualitas dari penjadwalan mata kuliah, sehingga suatu perkuliahan terjadwalkan dengan tidak melanggar *constraint* yang ditetapkan oleh pihak jurusan. *Constraint* tersebut terbagi dalam dua kategori yaitu *hard constraint* dan *soft constraint* (Petrovic, 2004).

Hard constraint merupakan *constraint* essensial yang harus terpenuhi dalam pembuatan jadwal, sedangkan *soft constraint* merupakan *constraint* nonessensial yang menentukan kualitas dari jadwal yang dihasilkan. *Hard constraint* dan *soft constraint* terjadi akibat interaksi antar komponen jadwal.

2.3 Algoritma Genetika

Algoritma genetika pertama kali ditemukan oleh John Holland pada tahun 1975melalui sebuah penelitian dari Universitas Michigan yang kemudian dikembangkan oleh muridnya David Goldberg pada tahun 1989. Awalnya tujuan Holland adalah bukan untuk merancang algoritma yang mampu memecahkan masalah yang spesifik, melainkan hanyauntuk mempelajari fenomena adaptasi seperti yang terjadi di alam sekaligus untuk mengembangkan cara-cara bagaimana mekanisme adaptasi alam dapat diimpor ke sebuahsistem computer (Hanita, 2011).

2.3.1 Definisi Algoritma Genetika

Algoritma Genetika merupakan salah satu metode untuk menentukan optimalisasi atas dasar Teori Darwin. Langkah prosedur algoritma ini diawali dengan menentukan suatu set solusi potensial dan melakukan perubahan dengan beberapa perulangan (iterasi) dengan algoritma genetika untuk menghasilkan solusi terbaik. Set solusi potensial ini ditetapkan diawal dan disebut dengan kromosom. Kromosom ini dibentuk secara random berupa susunan angka binary yang di-generate dan dipilih. Keseluruhan set dari kromosom yang diobservasi mewakili suatu populasi (Heli, dkk, 2010). Kromosom-kromosom berevolusi beberapa kali tahapan iterasi yang disebut dengan generasi. Generasi baru

(offsprings) degenerate dengan teknik kawin silang (crossover) dan mutasi (mutation). Crossover meliputi pemisahan atau pemecahan (splitting) dua kromosom, kemudian mengkombinasikan setengah bagian dari masing-masing kromosom dengan pasangan-pasangan lainnya. Mutasi meliputi pertukaran (flipping) satu bit (bagian) dari kromosom dengan satu bagian lain dari kromosom lain yang menjadi pasangannya. Kromosom-kromosom tersebut kemudian berevolusi dengan suatu kriteria kesesuaian (fitness) yang telah ditetapkan, hasil yang terbaik akan dipilih sedangkan yang lainnya diabaikan. Proses ini dilakukan secara berulang-ulang sampai menemukan suatu kromosom yang memiliki kesesuaian terbaik (best fitness) untuk dijadikan sebagai solusi terbaik dari suatu masalah (Suhartono, 2015).

Secara garis besar algoritma genetika memiliki banyak kesamaan dengan mekanisme genetika alami, baik dalam tahapan prosesnya maupun definisi dari istilah-istilah atau terminologi yang digunakan. Ekuivalensi istilah algoritma genetika dan genetika alami pada biologi dapat ditunjukkan pada Tabel 2.1.

Tabel 2.1 Perbandingan Istilah pada Genetika Alami dan Algoritma Genetika.

Genetika Alami	Algoritma Genetika
Kromosom	String
Gen	Fitur, Karakter atau Detector
Allel	Nilai Fitur
Locus	Posisi String
Genotip	Struktur
Fenotip	Set Parameter, Solusi Alternatif, Struktur yang Dikedokan

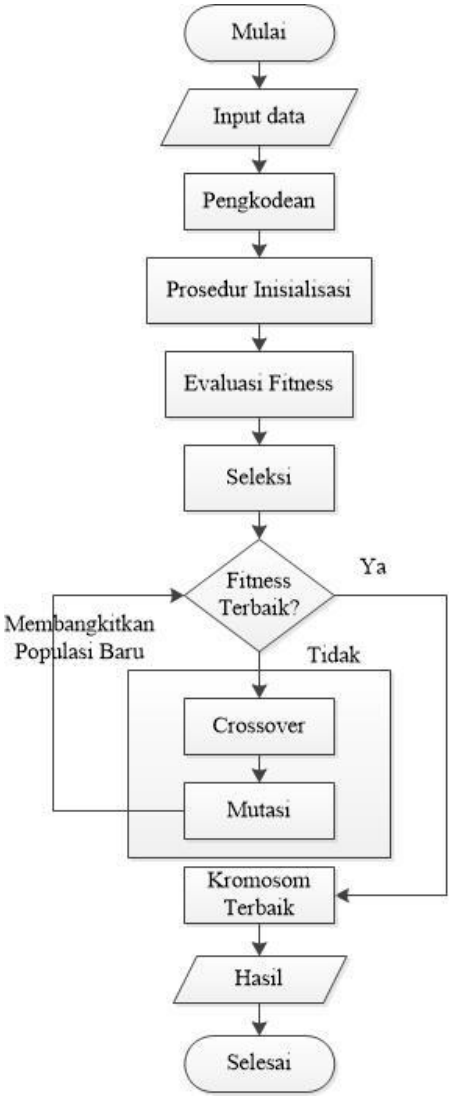
Algoritma Genetika merepresentasikan individu sebagai sebuah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Algoritma ini menyelesaikan permasalahan dalam pencarian kromosom yang terbaik. Kromosom-kromosom terbaik diperoleh dengan cara populasi awal yang dibangun secara acak dan populasi berikutnya yang merupakan hasil evolusi kromosom-kromosom melalui proses iterasi (seleksi, crossover dan mutasi) yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Setelah melalui beberapa generasi, maka akan diperoleh kromosom terbaik.

Beberapa definisi penting dalam algoritma genetika adalah sebagai berikut:

1) Gen adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini dapat berupa nilai biner, float, integer, kombinatorial maupun karakter.
2) Allel adalah nilai dari gen.
3) Kromosom adalah gabungan gen-gen yang membentuk nilai tertentu.
4) Individu menyatakan satu nilai atau menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
5) Populasi merupakan sekumpulan individu yang akan diproses Bersama dalam satu siklus proses evolusi.

6) Generasi menyatakan hasil evolusi kromosom-kromosom melalui iterasi.
7) Nilai *fitness* menyatakan seberapa baik nilai dari suatu individua tau solusi yang didapatkan.

Algoritma genetika secara umum dapat diilustrasikan pada flowchart berikut :



Gambar 2.2 Flowchart Algoritma Genetika

Dari Gambar 2.2 diatas, struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut :

1) Membangkitkan Populasi Awal

   Populasi awal dibangkitkan secara random, dimana populasi tersebut berisi beberapa kromosom yang telah didefinisikan sehingga didapatkan solusi awal. Populasi itu sendiri terdiri dari sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

2) Membentuk Generasi Baru

   Dalam membentuk generasi baru digunakan tiga operator yaitu operator reproduksi/seleksi, crossover dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi baru ini merupakan representasi dari solusi baru. Generasi baru ini dikenal denga istilah anak *(offspring)*.

3) Evaluasi Solusi

   Pada tiap generasi, kromosom akan diukur dengan fungsi *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai f*itness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Apabila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Beberapa kriteria berhenti yang sering digunakan antara lain:

(1) Berhenti pada generasi tertentu.

(2) Berhenti setelah beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi.

(3) Berhenti pada n generasi dimana nilai *fitness* dari populasi tidak mengalami perubahan.

2.3.2 Komponen-Komponen Utama Algortima Genetika

Ada 6 komponen utama yang terdapat di dalam algoritma genetika, (Kusumadewi : 2003) yaitu:

1) Teknik Pengkodean

   Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatukunci pokok persoalan ketika menggunakan algoritma genetika. Teknik pengkodean inimeliputi pengkodean gen dan kromosom. Gen merupakan bagian dari kromosom yangdapat direpresentasikan dalam bentuk string bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lain yang dapat diimplementasikanuntuk operator genetika.

2) Prosedur Inisialisasi

   Prosedur inisialisasi (membangkitkan populasi awal) adalah membangkitkansejumlah individu secara acak atau melalui prosedur tertentu. Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akandiimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus

dilakukaninisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasikromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domainsolusi dan kendala permasalahan yang ada.

3) Fungsi Evaluasi

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Dalam algoritma genetika, individu yang memiliki nilai *fitness* tinggi pada kromosomnya yang akan dipertahankan, sedangkan individu yang padakromosomnya bernilai *fitness* rendah akan diganti. Fungsi *fitness* tergantung pada permasalahan tertentu dari representasi yang digunakan.

Rumus *fitness* yang digunakan adalah sebagai berikut:

$$Fintess = \frac{1}{1+(F_1B_1+F_2B_2+\cdots)} \quad (2.1)$$

Keterangan : B_n = Bobot Pelanggaran

F_n = Banyaknya Pelanggaran

n = 1 … n

4) Seleksi

Tujuan dari proses seleksi untuk memilih individu-individu yang akan dipilih untuk proses persilangan dan mutasi, dimana diharapkan diperoleh calon induk yang baik. Induk yang baik akan menghasilkan keturunan yang baik. Langkah pertama dalam seleksi yaitu pencarian

nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiriterhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai f*itness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya.

Ada beberapa metode untuk memilih kromosom yang digunakan antara lain :

(1) Seleksi Roda Rolet (*Roulette Wheel Selection*)

Seleksi roda rolet dilakukan dengan cara menyeleksi parent dengan tujuan untuk mempertahankan nilai *fitness*-nya, dimana yang mempunyai kesempatan untuk diseleksi adalah kromoson yang baik. Pada metode ini dapat dibayangkan seperti sebuah permainan roda (roullette wheel), dimana semua kromosom ditempatkan dalam populasi, setiap tempat besar sesuai dengan fungsi *fitness*.

Kromosom dipilih berdasarkan nilai *fitness*, semakin besar nilai *fitness* maka kromosom tersebut mempunyai peluang untuk dipilih beberapa kali. Metode seleksi rodarolet merupakan metode yang paling sederhana, dan sering juga dikenal dengan nama *stochastic sampling with replacement*.

(2) Seleksi *Good Fitness*

Seleksi menggunakan metode *good fitness* dilakukan dengan cara setengah dari jumlah populasi yang memiliki harga *fitness* yang terendah akan dihilangkan, sehinggaakan hanya selalu tersisa

sekelompok solusi yang terbaik yang pernah diperoleh oleh program. Solusi yang tersisa hasil seleksi ini dikenal dengan nama populasi induk. Agar jumlah populasi tetap, maka perlu dibangkitkan solusi baru sebanyak setengah dari jumlah populasi yang ada. Dalam program penyeleksian, cara yang digunakan untuk membangkitkan solusi baru menggunakan dua cara yaitu reproduksi kromosom baru dancara mutasi dari solusi induk. Tujuan pembangkitan solusi baru ini untuk menemukanalternatif solusi yang lebih baik dari solusi-solusi yang sudah diperoleh.

5) Operator Genetika

Algoritma genetika merupakan proses pencarian yang heuristik dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Hal yang harusdiperhatikan adalah menghindari terjadinya konvergensi premature, yaitu mencapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalahhasil optimum lokal. Ada dua operator genetika yaitu:

(1) Persilangan (*Crossover*)

Salah satu komponen yang paling penting dalam algoritma genetika adalah persilangan (*crossover*). Persilangan merupakan proses pada algoritma genetika yang bekerja untuk menggabungkan dua kromosom orang tua (*parent*) menjadi kromosom baru (*offspring*) pada suatu waktu. Sebuah kromosom yang mengarah pada solusi

yang baik dapat diperoleh melalui proses *crossover* pada dua buah kromosom. Tidak semua kromosom mengalami persilangan. Cara sederhana pada proses persilangan yaitu denganmemilih suatu titik yang dipisahkan secara random dan kemudian membentuk *offspring*

dengan cara mengkombinasikan segmen dari satu induk ke sebelah kiri dari titik yang dipisahkan dengan segmen dari induk yang lain ke sebelah kanan dari titik yang dipisahkan.

Terdapat beberapa jenis *crossover*, antara lain :

(1) Crossover Satu Titik

Proses *crossover* dilakukan dengan memisahkan suatu string menjadi dua bagian, kemudian salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lainyang telah dipisahkan. Misalkan ada 2 kromosom dengan panjang 12 :

Induk 1 : 0 1 1 1 0 | **0 1 0 1 1 1 0**

Induk 2 : **1 1 0 1 0** | 0 0 0 1 1 0 1

Kromosom yang dipilih: 5

Anak 1 : 0 1 1 1 0 | 0 0 0 1 1 0 1

Anak 2 : **1 1 0 1 0 | 0 1 0 1 1 1 0**

(2) Crossover Banyak Titik

Proses *crossover* ini dilakukan dengan memisahkan suatu string menjadi beberapa bagian dan selanjutnya dipertukarkan dengan

bagian dari string yang lain yang telahdipisahkan. Misalkan ada 2 kromosom dengan panjang 12 :

Induk 1 : **0 1** | **1 1 0 0** | **1 0 1 1** | **1 0**

Induk 2 : 1 1 | 0 1 0 0 | 0 0 1 1 | 0 1

Posisi yang dipilih : kromosom setelah 2, 6 dan 10

Kromosom baru yang terbetuk:

Anak 1 : **0 1** | 0 1 0 0 | **1 0 1 1** | 0 1

Anak 2 : 1 1 | **1 1 0 0** | 0 0 1 1 | **1 0**

(2) Mutasi (*Mutaion*)

Operator berikutnya mutasi (*mutation*). Proses mutasi merupakan proses untuk mengubah salah satu atau lebih beberapa gen dari suatu kromosom. Proses ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Beberapa cara operasi mutasi yang diterapkan dalam algoritma genetika menurut jenis pengkodeannya, antara lain:

(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 (atau dengan menggunakan skema tertentu) pada kromosom. Contoh mutasi pada pengkodean biner:

Kromosom sebelum mutase : 1 0 0 1 0 **1** 1 1

Kromosom sesudah mutase : 1 0 0 1 0 **0** 1 1

(2) Mutasi Dalam Pengkodean Permutasi

Proses mutasi dalam pengkodean permutasi dilakukan dengan memilih dua posisi(lokus) dari kromosom dan nilainya saling dipertukarkan. Contoh mutasi dalam pengkodean permutasi :

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

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

6) Parameter Kontrol

Parameter kontrol genetika diperlukan untuk mengendalikan operator-operator seleksi. Pemilihan parameter genetika menentukan penampilan kinerja algoritma genetika dalam memecahkan masalah. Ada dua parameter dasar dari algoritma genetika, yaitu probabilitas *crossover* (pc) dan probabilitas mutasi (pm).

(1) Probabilitas *Crossover* (pc)

Probabilitas *crossover* akan mengendalikan operator *crossover* dalam setiapgenerasi dalam populasi yang mengalami *crossover*. Semakin besar nilai probabilitas *crossover,* akan semakin cepat struktur individu baru terbentuk ke dalam populasi. Sedangkan apabila nilai probabilitas *crossover* terlalu besar, individu yang merupakan kandidat solusi terbaik mungkin akan dapat hilang lebih cepat pada generasi selanjutnya. Disarankan nilai probabilitas *crossover* berkisar antara 80 % - 95 %.

(2) Probabilitas Mutasi (pm)

Probabilitas mutasi akan mengendalikan operator mutasi pada setiap generasidimana peluang mutasi yang digunakan biasanya lebih kecil daripada peluang *crossover*. Pada seleksi alam murni, mutasi jarang sekali muncul. Oleh karena itu, operator mutasi pada algoritma genetik juga tidak selalu terjadi. Untuk itulah nilai peluang mutasi dibuat lebih kecil untuk setiap generasi. Disarankan nilai probabilitas mutasi kecil berkisar antara 0.5 % - 1 %.

Parameter lain yang juga ikut menentukan efisiensi kinerja algoritma genetikaadalah ukuran populasi, yaitu banyaknya kromosom dalam satu populasi. Tidak adaaturan yang pasti tentang berapa nilai ukuran populasi. Apabila ukuran populasi kecil berarti hanya tersedia sedikit pilihan untuk *crossover* dan sebagian kecil dari domainsolusi saja yang dieksplorasi untuk setiap generasinya. Sedangkan apabila terlalu besar, kinerja algoritma genetik akan menurun. Penelitian menunjukan ukuran populasi besar tidak mempercepat pencarian solusi. Disarankan ukuran populasi berkisar antara 20–30.

2.4 Teori PHP (*Personal Home Page*)

PHP merupakan salah satu *server side scripting* (berjalan pada sisi server) yang populer saat ini. Selain dapat digunakan untuk berbagai sistem operasi, koneksi *database* yang sangat mudah, dan gratis menyebabkan PHP banyak digunakan. (Prasetyo, 2004) mengungkapkan bahwa PHP merupakan bahasa scripting server-side, dimana pemrosesan datanya dilakukan pada sisi server. Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada client yang melakukan permintaan.

Konsep kerja PHP diawali dengan permintaan suatu halaman web (file.php) oleh browser atau klien. Kemudian berdasarkan alamat di Internet (URL), *browser* mendapatkan alamat dari *web server*, yang akan mengidentifikasi halaman yang diminta, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, ketika file PHP yang diminta didapatkan oleh *web server*, isinya segera dikirimkan ke mesin PHP untuk diproses dan memberikan hasilnya (berupa kode HTML) ke *web server*, lalu menyampaikannya ke klien.