

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Fondasi Telapak

Fondasi Telapak (*Foot Plate Foundation*) adalah fondasi bangunan tingkat rendah-menengah yang dibuat dengan beton bertulang. Fondasi Telapak meneruskan beban ke tanah dengan prinsip membagi beban dari kolom yang besar dengan luasan fondasi yang besar sehingga tegangan yang terbentuk dapat ditampung dengan tegangan ijin tanah.

Menurut Peck *et al.* (1973), fondasi telapak adalah pelebaran alas kolom atau dinding dengan tujuan untuk meneruskan beban pada tanah dasar pada suatu tekanan yang sesuai dengan sifat-sifat tanah yang bersangkutan. Telapak yang mendukung kolom tunggal disebut telapak kolom individual, telapak tersendiri, atau telapak sebar. Telapak di bawah suatu dinding disebut telapak dinding atau telapak menerus. Sebuah telapak mendukung beberapa kolom dinamakan telapak gabungan.

#### 2.2 Optimasi Teknik

Optimasi Teknik (*engineering optimization*) adalah teknik optimasi dalam suatu permasalahan teknik. Teknik optimasi yang dimaksud di sini adalah teknik untuk menemukan solusi yang paling efektif dan efisien terhadap variabel yang dioptimalkan dari suatu permasalahan yang ada.

Menurut Yang (2018), Banyak permasalahan optimasi dalam sains dan ilmu teknik berbentuk non-linier. Permasalahan ini membutuhkan teknik optimasi

optimasi yang sangat baik untuk dipecahkan. Teknik tradisional seperti algoritma yang berdasar pada gradien (*gradient-based algorithms*) hanya dapat menemukan optimum lokal, dan sering gagal dalam menemukan solusi optimum global. Sehingga belakangan ini dicari algoritma yang baik dan algoritma yang terinspirasi dari alam sering digunakan. Salah satu contohnya adalah Algoritma Genetik.

### 2.3 Algoritma Genetik

Algoritma Genetik (*Genetic Algorithm*) adalah algoritma optimasi yang terinspirasi dari teori evolusi yang dikemukakan Charles Darwin. Algoritma ini menggunakan sistem seleksi alam (*natural selection*) yang memiliki prinsip individu yang paling baik dipilih untuk menciptakan generasi penerus yang lebih baik.

Menurut Arfiadi (2016), Algoritma Genetik telah banyak digunakan untuk menyelesaikan permasalahan optimasi. Salah satu kelebihan Algoritma Genetik adalah dapat menyelesaikan permasalahan optimasi yang kompleks dengan cara yang mudah. Dengan Algoritma Genetik, nilai yang relatif dekat dengan optimum global dapat dicapai dengan mudah.

Menurut Arfiadi *et al.* (2001) Algoritma Genetik memiliki kemampuan untuk menemukan solusi optimum pada domain yang tidak diketahui. Selain itu, Algoritma Genetik juga merupakan algoritma yang tidak menggunakan gradien dalam proses optimasinya sehingga penggunaannya menjadi mudah.

Menurut Yang (2014), Algoritma Genetik adalah salah satu algoritma yang sangat terkenal dan evolusioner dalam hal pengaplikasian. Banyak permasalahan

optimasi yang terkenal dapat diselesaikan dengan Algoritma Genetik. Banyak algoritma evolusioner modern yang dibuat berdasarkan Algoritma Genetik dan memiliki kesamaan yang dengan Algoritma Genetik.

*Genetic Algorithm* memiliki proses sebagai berikut:

1. pembuatan populasi awal (*intial population*),
2. perhitungan fungsi *fitness* (*fitness function*),
3. seleksi (*selection*), penyilangan (*crossover*) dan mutasi (*mutation*), dan
4. penghentian repetisi (*termination*).

Proses tersebut digambarkan dalam *flowchart* pada lampiran.

### **2.3.1 Pembuatan populasi awal (*initial population*)**

Individu-individu dibuat sejumlah populasi. Jumlah populasi bergantung pada kompleksitas permasalahan yang akan dioptimasi. Semakin tinggi kompleksitas maka jumlah populasi yang dibuat sebaiknya semakin banyak. Individu-individu awal ini dapat dibuat secara acak maupun terkonsentrasi pada daerah yang memiliki probabilitas tinggi untuk menghasilkan *fitness* lebih tinggi.

### **2.3.2 Perhitungan fungsi *fitness* (*fitness function*)**

Fungsi *fitness* menentukan seberapa baik sebuah individu (kemampuan individu untuk bersaing dengan individu lain). Fungsi *fitness* ini memberikan skor *fitness* kepada setiap individu. Probabilitas seorang individu berproduksi untuk generasi berikutnya ditentukan berdasarkan skor *fitness* ini.

### **2.3.3 Seleksi (*selection*), penyilangan (*crossover*), dan mutasi (*mutation*)**

Ide dari seleksi (*selection*) adalah memilih individu dengan skor *fitness* terbaik dan meneruskan gen mereka ke generasi yang berikutnya. Sepasang

individu dipilih akan dipilih berdasarkan nilai *fitness* mereka. Individu dengan nilai *fitness* yang lebih baik memiliki peluang lebih tinggi untuk mengalami seleksi. Setelah itu terbentuklah anakan (*offspring*) yang dapat mengalami penyilangan dan mutasi.

Penyilangan (*crossover*) adalah bagian yang sangat penting dalam Algoritma Genetik. Untuk sepasang individu yang akan menjadi orang tua (*parents*), sebagian gennya akan disilangkan untuk menghasilkan individu yang sedikit berbeda dengan orang tuanya.

Untuk setiap individu yang dibentuk, sebagian dari mereka akan mengalami mutasi (*mutation*). Mutasi membuat individu yang terbentuk memiliki gen yang berbeda jauh dengan orang tuanya. Tujuan dari mutasi ini adalah untuk memperluas jangkauan dalam hal mencari optimum global.

Karena jumlah individu dalam populasi Algoritma Genetik tidak berubah, maka individu-individu dengan nilai *fitness* yang buruk akan mati dan tergantikan dengan individu baru hasil perkawinan (*offspring*) dari proses penyilangan dan mutasi di atas. Ada juga individu baru yang kembali dibuat dengan proses insialisasi untuk tetap menjaga keberagaman (*diversity*) individu-individu agar memudahkan proses optimasi menghindari optimum lokal.

#### **2.3.4 Penghentian repetisi (*termination*)**

Repetisi berhenti ketika algoritma program telah mencapai suatu tujuan yang telah ditetapkan. Misalnya ketika jumlah generasi telah terpenuhi ataupun hasil yang telah konvergen. Hasil yang telah konvergen inilah yang disebut sebagai solusi optimal dari proses *Genetic Algorithm* ini.