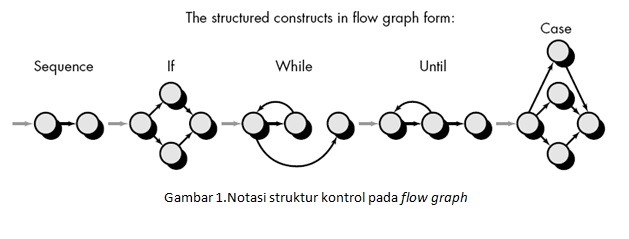
*Cyclomatic Complexity* adalah sebuah *software metric* yang menyediakan ukuran kuantitatif dari kompleksitas logika dari sebuah program. Dengan menggunakan hasil pengukuran atau perhitungan dari *metric cyclomatic complexity*, kita dapat menentukan apakah sebuah program merupakan program yang sederhana atau kompleks berdasarkan logika yang diterapkan pada program tersebut. Apabila dikaitkan dengan pengujian perangkat lunak (*software testing*), *cyclomatic complexity* dapat digunakan untuk menentukan berapa minimal *test case*yang harus dijalankan untuk menguji sebuah program dengan menggunakan teknik *basis path testing*. Pada pengujian *basis path,*aliran control logika digambarkan dengan menggunakan *flow graph*. Berikut ini adalah notasi struktur kontrol pada *flow graph*untuk menggambarkan sekuensial, seleksi, maupun perulangan:



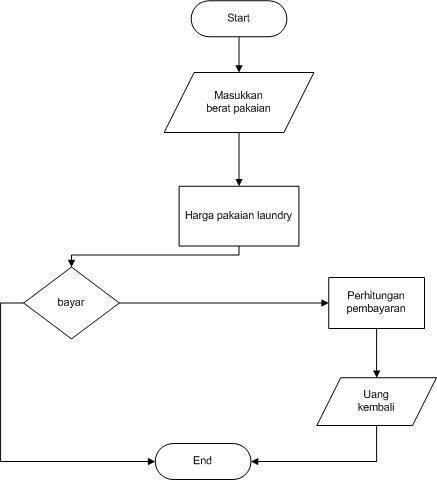
Notasi lingkaran disebut sebagai *flow graph node* yang digunakan untuk menggambarkan *statement-statement*berikut:

1. Satu atau lebih statement secara sekuensial yang dikelompokkan
2. Percabangan seleksi dari satu statement kedua pilihan statement (seleksi)
3. Penggabungan dua statement yang dilanjutkan pada satu statement yang sama (merge)

Sedangkan notasi garis panah disebut sebagai *edge*atau *link*, menggambarkan aliran kontrol. Setiap *edge*harus dihubungkan dari/ke sebuah *node*, meskipun *node*tersebut tidak mewakili sebuah *statement*khusus. Area yang dibatasi oleh *node* dan *edge*disebut sebagai *region*

Kelas Customer

1. Pembayaran



Dari flow graph yang sudahtersedia, cyclomatic complexity dari sebuah program dapat dibuat dengan menggunakan rumus dibawah ini:

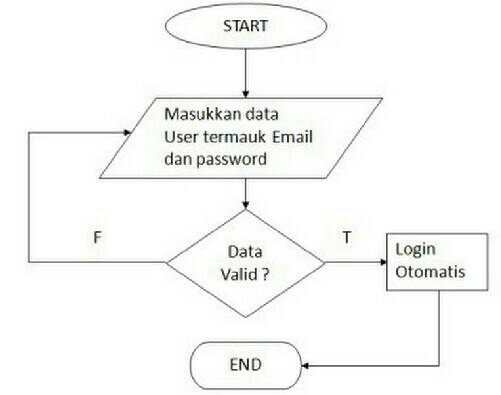


V(G)     : cyclomati ccomplexity  
E          : total jumlah edge  
N          : total jumlah node

Maka :

V(G) = 7 – 7 + 2 = 2

1. Login



Dari flow graph yang sudahtersedia, cyclomatic complexity dari sebuah program dapat dibuat dengan menggunakan rumus dibawah ini:



V(G)     : cyclomati ccomplexity  
E          : total jumlah edge  
N          : total jumlah node

Maka :

V(G) = 5 – 5 + 2 = 2