**DRAFT PROPOSAL**

**Judul :**

**Algoritma perancangan perangkat lunak pada bahasa pemrogramman C++ yang bertujuan untuk mencegah terjadinya memory leak**

**Identitas :**

Nama : Mahfudh Afandi

NIM : 1603015041

Semester : VI / Genap

1. **Pendahuluan**

**1.1 Latar Belakang**

Saat ini, komputer telah menjadi salah satu kebutuhan utama khususnya bagi mereka yang memang bekerja di bidang yang selalu bersentuhan dengan komputer seperti pekerja kantoran, PNS, Designer, programmer, mahasiswa dan masih banyak lagi pekerjaan yang memang membutuhkan perangkat komputer. Dimana sebagian besar dari perkerjaan yang menggunakan komputer tersebut pastinya juga dibantu dengan adanya keberadaan perangkat lunak didalam komputer yang digunakan, agar dapat mempermudah proses interaksi antara komputer dengan pengguna sehingga dapat menghasilkan output secara mudah.

**1.2 Rumusan Masalah**

Namun, terkadang ketika sedang menggunakan suatu perangkat lunak pada komputer yang biasanya dipakai dalam bekerja tersebut, sering sekali mengalami masalah terkait dengan performa yang terkadang sangat lambat atau bahkan not responding sehingga menghambat pekerjaan dikarenakan perangkat lunak komputer yang tidak bekerja secara maksimal. Mungkin kebanyakan dari orang yang terkena masalah ini kurang mengetahui penyebab utamanya secara spesifik sehingga menimbulkan keputusan pribadi yang belum tentu benar dan bahkan dapat memperburuk keadaan.

**1.3 Batasan Masalah**

Dari sebagian besar penyebab yang timbul ketika perangkat lunak mengalami masalah lambat atau not responding ialah salah satunya dikarenakan oleh kejadian yang disebut sebagai "Memory leaks" *(Kebocoran Memori)*. kejadian ini terjadi dikarenakan adanya kesalahan algoritma pada saat perangkat lunak dibuat, dimana kesalahan tersebut mengakibatkan pemakaian memori yang terpakai terlalu banyak secara terus menerus disaat perangkat lunak dijalankan sampai dimana kondisi memori *(RAM)* tidak dapat menampungnya lagi, dan disaat itulah memori mencapai batasnya lalu menyebabkan perangkat lunak tersebut akan berhenti bekerja.

Selanjutnya agar pembahasan mengenai masalah memory leaks ini dapat dimengerti dengan jelas, maka pembahasan masalahnya dilakukan didalam lingkungan bahasa pemrogramman C++.

**1.4 Manfaat dan Tujuan**

Tentunya, manfaat yang dapat dihasilkan dari algoritma ini yaitu agar membuat perangkat lunak yang telah dirancang untuk tidak memakan terlalu banyak memori ketika sedang dijalankan, dan meningkatkan performa pada komputer agar terasa ringan ketika digunakan untuk menjalankan perangkat lunak tersebut walaupun dengan kondisi spesifikasi perangkat keras komputer yang standar.

Kemudian tujuan utama dengan dibuatnya algoritma ini ialah agar dapat meminimalisir terjadinya *Memory leaks* pada suatu perangkat lunak komputer, serta meningkatkan teknik dalam perancangan suatu perangkat lunak agar menjadi lebih baik dan effesien bagi setiap pengembang kedepannya.

.

1. **Landasan Teori**

Memori (atau lebih tepat disebut sebagai memori fisik) merupakan istilah generik yang merujuk pada media penyimpanan data sementara pada komputer. Setiap program dan data yang sedang diproses oleh prosesor akan disimpan di dalam memori fisik. Data yang disimpan dalam memori fisik bersifat sementara, karena data yang disimpan di dalamnya akan tersimpan selama komputer tersebut masih dialiri daya. Memori fisik umumnya diimplementasikan dalam bentuk Random Access Memory (RAM) yang bersifat dinamis (DRAM).

Memory Leak adalah suatu keadaan dimana resource dari aplikasi yang dialokasikan pada memori tidak mampu dibebaskan sehingga membebani memori dan pada akhirnya menurunkan performa atau dalam kasus terburuknya menyebabkan sistem crash pada suatu perangkat lunak komputer.

**2.1 Penyebab Terjadinya Memory Leak**

Memory leaks terjadi ketika memori baru di alokasikan namun tidak pernah di de-alokasikan (dihapus/dipindahkan). Pada program C, memori baru dialokasikan oleh fungsi *malloc* atau *calloc*, dan di de-alokasikan oleh fungsi *free*. Lalu pada C++, memori baru biasanya dialokasikan oleh Operator *new* dan di de-alokasikan oleh Operator *delete* atau *delete[ ].* Masalah dengan adanya memory leaks disebabkan karena hal tersebut menumpuk dari waktu ke waktu, dan jika dibiarkan dapat membuat program menjadi lambat atau bahkan membuat program menjadi crash.

Salah satu kesalahan paling umum yang menyebabkan terjadinya memori leaks adalah dengan menggunakan operator *delete* berdasarkan penerapan yang salah. Operator *delete* harus digunakan untuk membersihkan sebuah class yang dialokasikan atau menghapus data nilai. Sedangkan Operator *delete[]* seharusnya digunakan untuk membersihkan data nilai dalam bentuk array.

1. **Metodologi**
   1. **Mendeteksi adanya Memory Leak**
   2. **Menangani Memory Leak**
   3. **Tahap penyelesaian**