# BAB I
# PENDAHULUAN

## I.1 Latar Belakang Masalah

*Self-Modifying Code* berperan penting dalam pengembangan perangkat lunak komersil terutama pada kasus proteksi lisensi dari pembajakan dan *malicious software* dalam menyembunyikan eksistensinya. SMC juga sering dikombinasikan dengan teknik *anti-debugging* lainnya seperti *obfuscation.* Hal tersebut menjadi tantangan pada dunia *reverse engineering* karena sulit dianalis secara statis. Aplikasi debugger umumnya hanya mengontrol data tetap pada program dan memori saja. Salah satu menyiasatinya adalah dengan mendampingi *debugger* dengan metode instrumentasi. Instrumentasi pada pemograman digunakan untuk beberapa tujuan seperti hasil penggunaan memori, statistik performansi maupun pengecekan *bug* dari segi kegunaan dan keamananan.

Instrumentasi beroperasi dengan menanamkan kode *monitoring* ke dalam spesifik komponen program pada level sumber kode maupun biner. Pada kasus program *Self-Modifying Code* terdapat pola subtitusi fungsi yang membuat proses *monitoring* lepas dari pemindaian. Hal tersebut merupakan kelemahan dari instrumentasi statis yang bekerja sebelum program dijalankan atau hanya memindai binernya saja. Dengan kata lain subtitusi fungsi tidak dapat diketahui karena terjadi pada saat eksekusi program. Kesulitan lain yang akan dihadapi oleh *reverse engineer* adalah waktu yang diperlukan untuk memecahkan subtitusi tersebut dan akan selalu berganti setiap kali ditemukan polanya, maka dari itu perubahan harus dapat ditebak akan seperti apa selanjutnya.

Berdasarkan kendala tersebut diperlukan suatu analisis secara dinamis pada saat program ditahap *runtime.* Dengan menggunakan metode *Dynamic Binary Instrumentation* diharapkan dapat menganalisis *Self-Modifying Code* saat program sedang memodifikasi dirinya. Instrumentasi dinamis pada biner juga menjaga integritas biner sehingga instruksi program tetap utuh atau tidak menganggu

proses subtitusi fungsi dari *Self-modifying Code* itu sendiri. Proses DBI dilakukan hampir tanpa interaksi *reverse engineer* atau menganalisis secara otomatis, dengan begitu kasus dapat cepat diselesaikan.

## I.2 Perumusan Masalah

Berdasarkan permasalahan yang telah dikemukakan pada latar belakang dapat dirumuskan, kesulitan melakukan *reverse engineering* terhadap program *Self-Modifying Code* sebagai usaha *anti debugging*.

## I.3 Maksud dan Tujuan Penelitian

Maksud dari penelitian ini adalah menganalisis teknik *Self-Modifying Code* menggunakan metode *DBI(Dynamic Binary Instrumentation).*

Adapun tujuan yang akan dicapai dalam penelitian ini adalah memperoleh hasil analisis *Self-Modifying Code* yang mudah dimengerti dengan keluaran *intermediate representation* untuk mempermudah *reverse engineer* dalam menilai tindak lanjut pada proses *debugging.*

## I.4 Batasan Masalah

Batasan masalah yang digunakan untuk *Dynamic Binary Instrumentation* adalah sebagai berikut:

1. *Dynamic Binary Instrumentation* menggunakan *Open Source Framework DynamoRIO.*
2. *DynamoRIO* digunakan dengan integrasi API (*Application Programming Interface)* LLVM untuk memperoleh keluaran *intermediate representation*
3. *DynamoRIO plugin* yang dibangun bertujuan memberikan nasihat atas pola *Self-Modifying Code*
4. *DynamoRIO plugin* yang dibangun bukan sebagai *patcher* atau mengubah program *Self-Modifying Code* menjadi *non-Self-Modifying Code*
5. *DynamoRIO plugin* dibangun menggunakan bahasa pemograman C++ sebagaimana API yang disediakan oleh *DynamoRIO*

Sedangkan batasan masalah pada kasus *Self-Modifying Code* terdiri dari:

1. Program *Self-Modifying Code* berformat ELF (Executable and Linkable Format)
2. Program *Self-Modifying Code* mempunyai instruksi arsitektur x86_64 atau AMD64(*little endian)*
3. Program *Self-Modifying Code* dianalisis dengan metode *Dynamic Binary Instrumentation*
4. Program *Self-Modifying Code* dianalisis dengan tujuan menemukan set instruksi pengganti dengan mudah
5. Program *Self-Modifying Code* bukan kompilasi JIT(*Just-in-Time)*

## I.5 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini adalah metode eksperimental karena dilakukan pengujian berulang – ulang dan kasus dapat diuji dengan metode lainnya. Adapun untuk analisis permasalahan digunakan metode *DBI(Dynamic Binary Instrumentation).*

1. Metode Pengumpulan Data

a. Metode Studi Pustaka

Mencari literatur yang terkait dengan *Self-Modifying Code* dan *Dynamic Binary Instrumentation*.

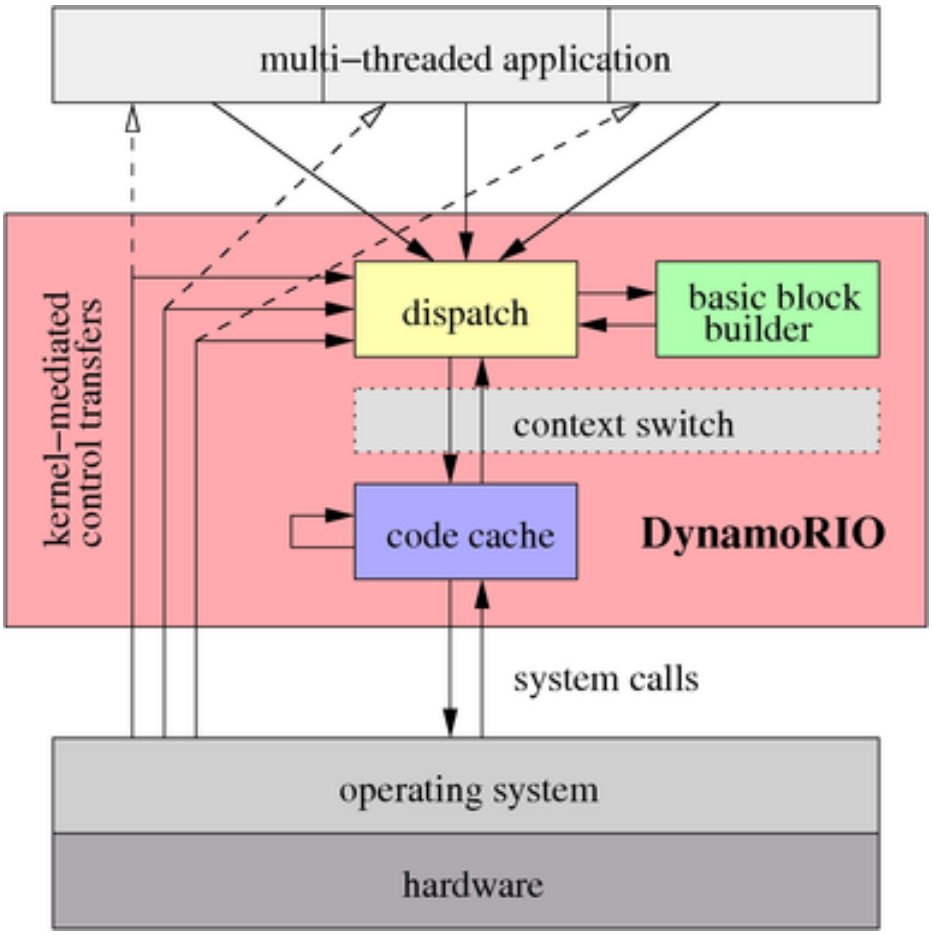
b. Metode Wawancara

Berupa tanya jawab dengan *reverse engineer* melalui *e-mail* dan forum *online*. Metode ini digunakan untuk mendapat data – data pada sistem.

2. Metode Analisis Dynamic Binary Instrumentation

Metode yang digunakan dalam penelitian ini adalah *DBI(Dynamic Binary Instrumentation)* dengan menggunakan *framework* dari *DynamoRIO*. *DynamoRIO* beroperasi dengan menggeser eksekusi sebuah aplikasi dari petunjuk aslinya ke sebuah *cache code*, dimana instruksi dapat bebas dimodifikasi. *DynamoRIO*

menempati ruang alamat aplikasi tersebut dan memiliki kendali penuh atas eksekusi, mengambil alih setiap kali kontrol meninggalkan *cache code* atau ketika sistem operasi langsung mentransfer kontrol ke aplikasi (*kernel-mediated control transfers*).




Gambar I.1 Proses DynamoRIO

*DynamoRIO* menyalin kode aplikasi satu blok dasar dinamis setiap waktu ke dalam kode *cache* blok dasarnya. Sebuah blok yang langsung menargetkan blok lain sudah ditempati dalam *cache* yang terkait blok tersebut untuk menghindari kembali ke *dispatcher*. Urutan eksekusi dari blok dasar yang sering muncul digabungkan menjadi *traces* yang ditempatkan secara terpisah. *DynamoRIO* membuat *traces* tersebut tersedia melalui antarmuka untuk kemudahan akses ke kode *stream*.

## I.6 Sistematika Penulisan

Sistematika penulisan ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Menguraikan tentang latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan tujuan dan kegunaan penelitian, yang kemudian diikuti dengan pembatasan masalah, asumsi, serta sistematika penulisan.

BAB II. TINJAUAN PUSTAKA

Membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang telah pernah dilakukan sebelumnya termasuk sintesisnya.

BAB III. ANALISIS SELF-MODIFYING CODE DAN PERANCANGAN DYNAMIC BINARY INSTRUMENTATION

Bab ini berisi tentang analisis masalah, analisis biner program, analisis non-fungsionalitas serta analisis perancangan metode yang digunakan. Selain itu pada bab ini memaparkan perancangan sistem yang akan dianalisis.

BAB IV. IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini menerangkan tentang pengujian sistem berdasarkan pada metode yang digunakan dengan menggunakan *unit testing* dan menjelaskan apakah sudah benar-benar sesuai dengan analisis dan perancangan yang telah dilakukan.

BAB V. KESIMPULAN DAN SARAN

Berisi kesimpulan dan saran yang sudah diperoleh dari hasil penulisan penelitian.