Write up soal pengisian

Pertama mari kita cek jenis filenya :

```
kazuya@ubuntu:~/Desktop/codingan$ file soal_pengisian
soal_pengisian: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamica
lly linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildI
D[sha1]=c83a5c3373bc116079c2eaa45b88c7766aa23b8c, not stripped
kazuya@ubuntu:~/Desktop/codingan$
```

Soal pengisian ini ternyata berjenis ELF 64-bit kemudian mari kita coba untuk menjalankan program ini untuk melihat aktivitas apa yang dilakukan program ini

```
kazuya@ubuntu:~/Desktop/codingan$ ./soal_pengisian
Sabardong
 Halo reverse aku dong
coba cek rahasianya
Sabardong
 Halo reverse aku dong
coba cek rahasianya
Sabardong
 Halo reverse aku dong
coba cek rahasianya
Sabardong
 Halo reverse aku dong
coba cek rahasianya
Sabardong
 Halo reverse aku dong
coba cek rahasianya
```

Program ini melakukan loopingan yang memprint beberapa teks dan kita tidak mengetahui kapan program ini akan selesai sehingga jalan terbaik adalah dengan mereverse program ini sebenernya disini saya akan mencoba melakukan ini dengan 2 cara pertama dengan IDA pro dan Strings untuk mengecek string apa saja yang akan di print oleh program ini :

```
kazuya@ubuntu:~/Desktop/codingan$ strings soal_pengisian
/lib64/ld-linux-x86-64.so.2
:\3s
libc.so.6
puts
printf
getchar
usleep
__cxa_finalize
__libc_start_main
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
=9
AWAVA
AUATL
[]A\A]A^A_
Sabardong
 Halo reverse aku dong
coba cek rahasianya
flag=basic_reverse_bisa_
gak_liat_sampe_sini_!
```

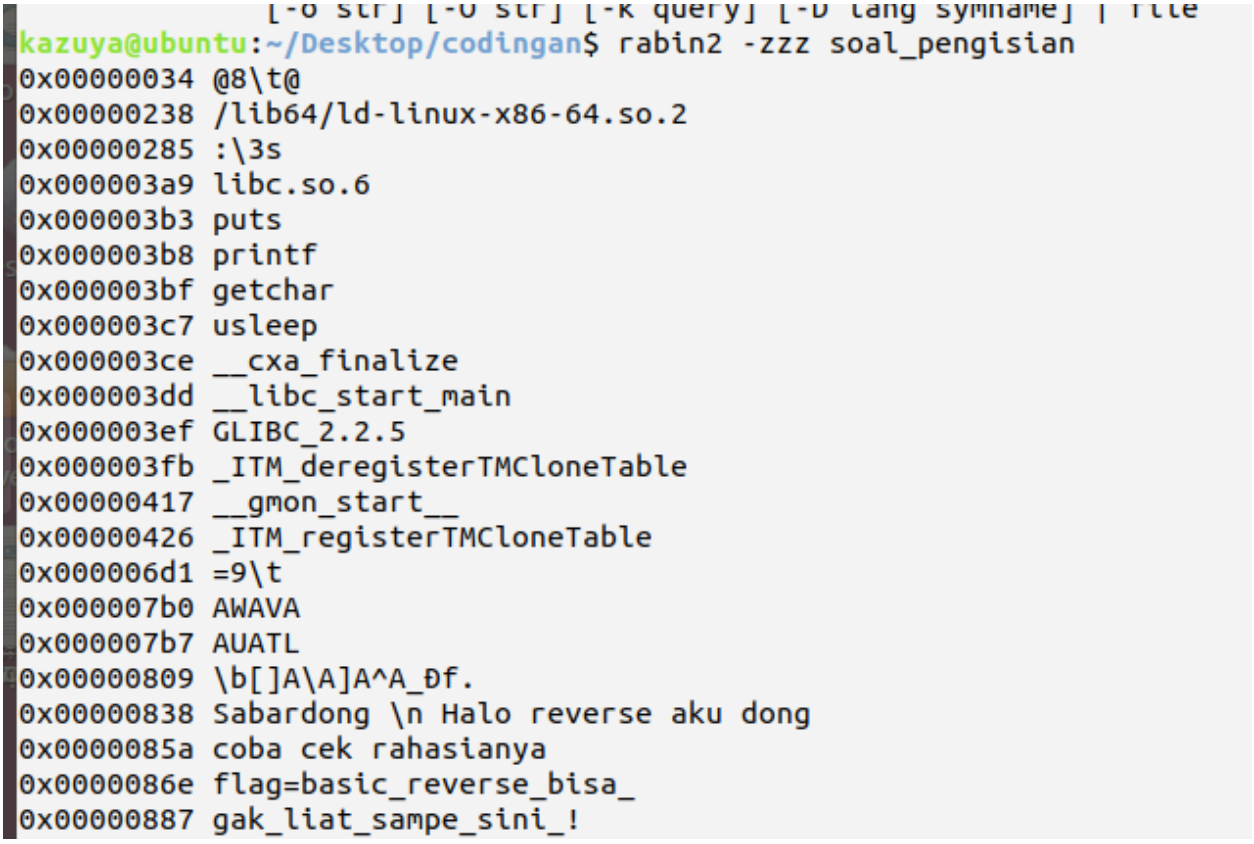
Ketika kita melakukan nya dengan string kita dapat melihat ada 5 string yang dapat di print oleh program ini. Kemudian jika kita melakukan dengan melakukan IDA pro

Akan terlihat seperti ini

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  signed int i; // [rsp+Ch] [rbp-4h]

  for ( i = 0; i <= 9999999; ++i )
  {
    print(20);
    puts("Sabardong \n Halo reverse aku dong");
    puts("coba cek rahasianya");
  }
  printf("flag=basic_reverse_bisa_", argv, envp);
  puts("gak_liat_sampe_sini_!");
  getchar();
  return 0;
}
```

Nah ini adalah function main yang terdapat di program tersebut kita dapat melihat bahwa program ini menjalankan 3 string awal dan di print sampai 9999999999 kemudian setelah selesai makan dia baru akan memprint si flag nya, namun masih ada 1 cara lagi yaitu dengan menggunakkan



Yaitu dengan menggunakkan rabin2 –zzz untuk memprint semua string yang terdapat didalam program tersebut atau bisa kita bilang hanya string yang dapat di print dan masih berbentuk kata-kata.

Jadi flagnya adalah BeeFest{basic_reverse _bisa_gak_liat_sampe_sini_!}

Tujuan dari reverse ini adalah dengan menunjukkan bahwa kita dapat melihat string yang di print di seluruh program ini saat program sedang dijalankan.