Fast Exponent

Cryptography

Nama: Ramadhan Kalih Sewu

NPM: 1806148826

Fungsi eksponen dibawah akan dibandingkan kompleksitas waktu algoritma-nya.

```
function [y] = RegularExponent(base, exp, modulo)
    if (exp == 0) y = 1; else y = base; end
    for i = 1:exp
        y = y * y;
    end
    y = mod(y, modulo);
end
```

```
function [y] = FastExponent(base, exp, modulo)
    binvec = de2bi(exp);
    y = 1; a = base;
    for i = 1:length(binvec)
        if (binvec(i)) y = mod(y * a, modulo); end
        a = mod(a^2, modulo);
    end
end
```

Kita tentukan nilai satuan dan pangkat berdasarkan jumlah digitnya. Semakin besar maka tentunya akan memperlambat jalannya program. Pangkat adalah faktor terbesar dari kompleksitas waktu, ini dilihat dari algoritma diatas. Sedangkan nilai modulo di atur dengan nilai yang tetap. Setelah saya coba dengan digit desimal bervariasi, nilai modulo tidak terlalu berpengaruh terhadap kompleksitas jalannya program.

```
listBase = [15, 262, 2515, 92124, 219124, 7823112, 81231231];
listExp = [91, 231, 5122, 12832, 341243, 3217312, 73312122];
fixMod = 21;
```

Menentukan kebenaran program (Unit Test):

```
count = 0;
if (FastExponent(2, 50, 13) == 4) count = count + 1; end
if (FastExponent(2, 40, 13) == 3) count = count + 1; end
if (FastExponent(2, 90, 13) == 12) count = count + 1; end
if (FastExponent(17, 22, 21) == 4) count = count + 1; end
sprintf('%d/4', count)
```

```
>> Source

ans =

4/4
```

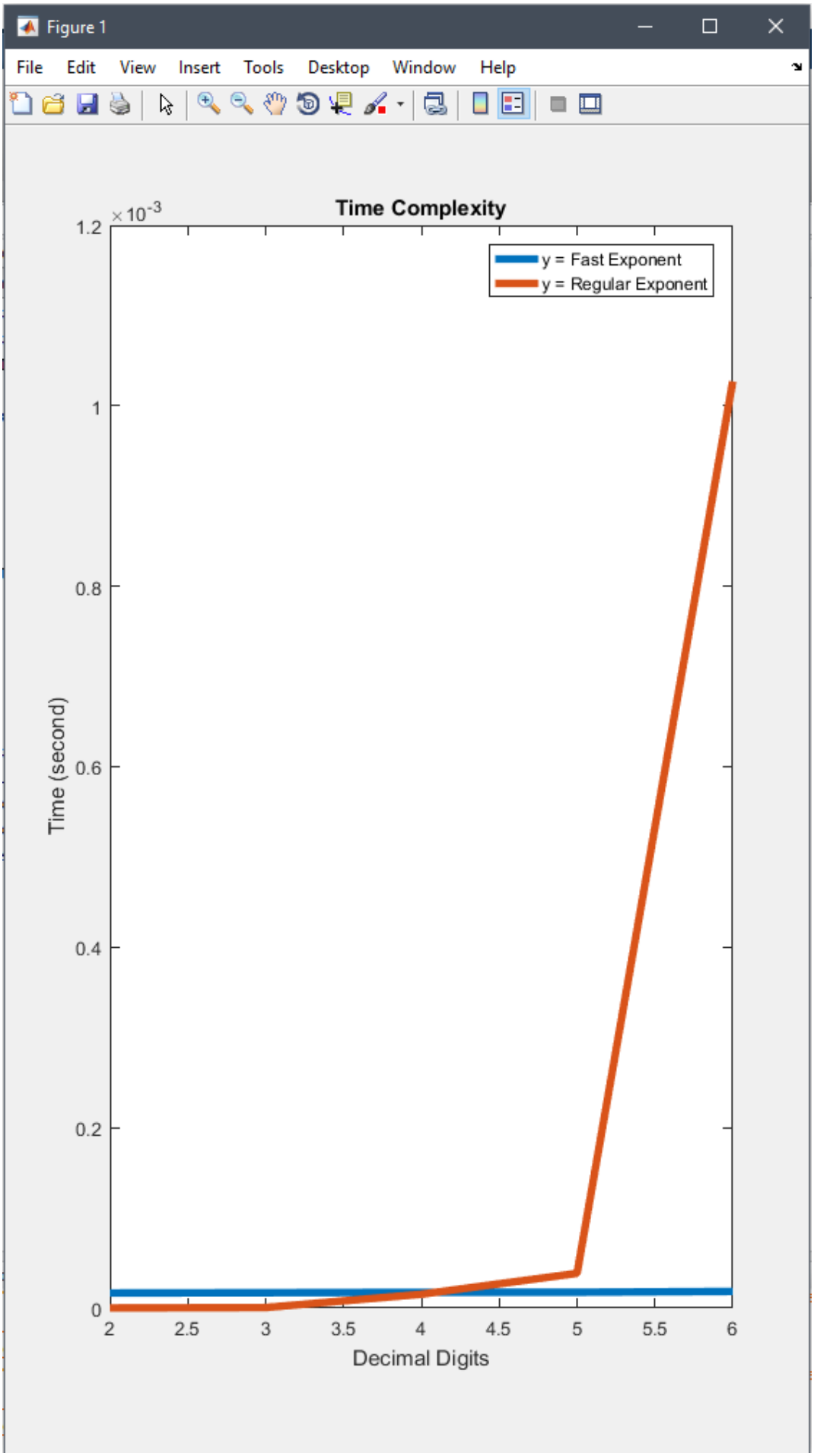
Figure 1
File Edit View Insert Tools Desktop Window Help
Time Complexity
×10^-3
y = Fast Exponent
y = Regular Exponent
Time (second)
Decimal Digits
0
0.2
0.4
0.6
0.8
1
1.2
2
2.5
3
3.5
4
4.5
5
5.5
6

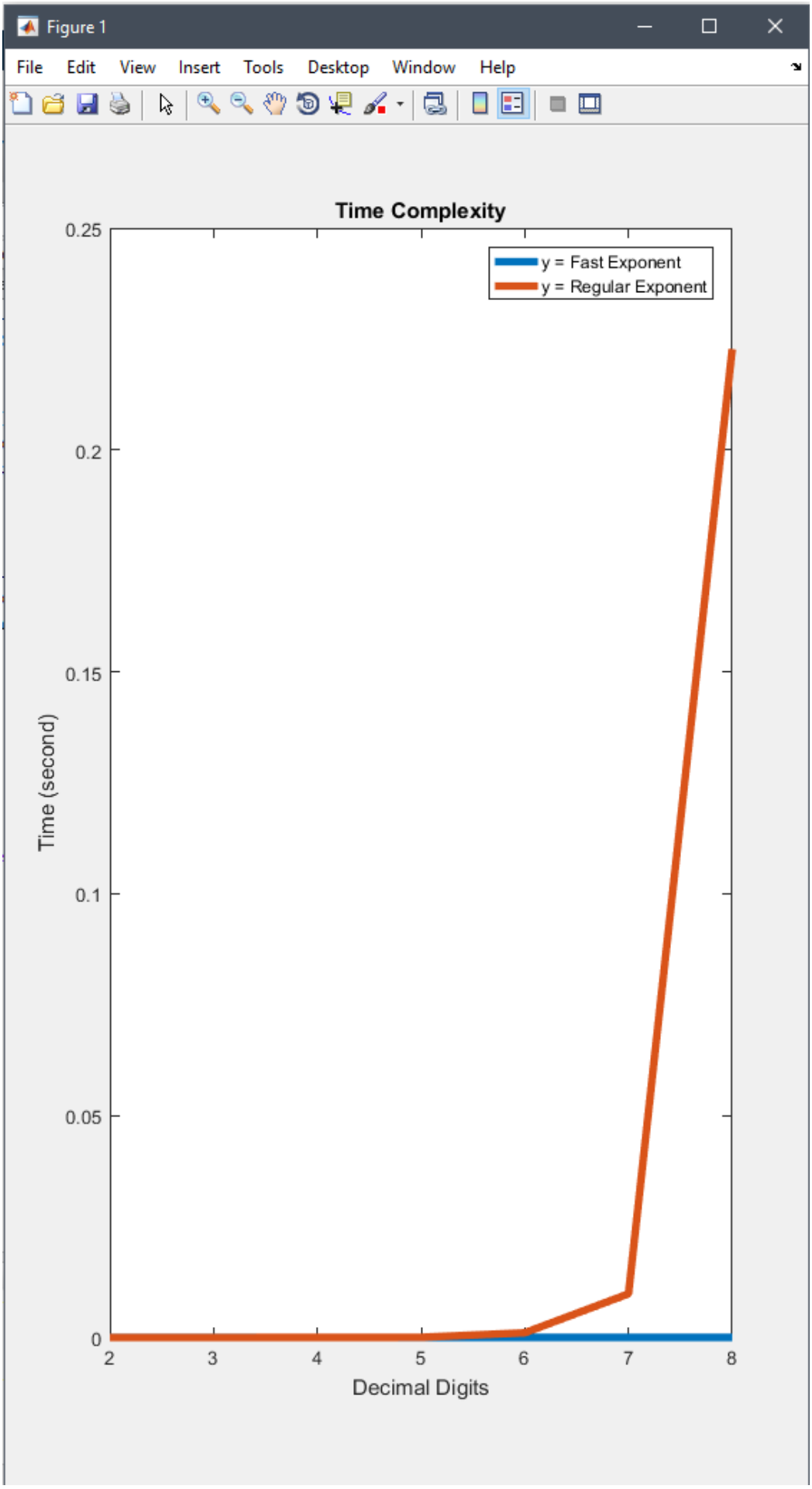
Figure 1
File Edit View Insert Tools Desktop Window Help
Time Complexity
y = Fast Exponent
y = Regular Exponent
0.25
0.2
0.15
0.1
0.05
0
Time (second)
2 3 4 5 6 7 8
Decimal Digits

Dilihat dari grafik, kita dapat lihat bahwa kompleksitas waktu algoritma Fast Exponent lebih baik dibandingkan dengan fungsi eksponen pada umumnya. Pada pangkat dengan digit desimal kurang dari empat, fungsi Fast Exponent bekerja lebih buruk, tetapi kenaikannya sangat minimal dibanding dengan fungsi eksponen umumnya. Kenapa berat di awal?

- fungsi yang dijalankan pada inisiasi menentukan digit biner.
- fungsi iterasi yang lebih berat

Kompleksitas waktu ini dapat dengan mudah kita analisa dalam kode program dimana pada fungsi eksponen reguler, memiliki:

$$O(n)$$

Sedangkan pada fungsi fast eksponen, memiliki kompleksitas waktu:

$$O(\log_2 n + 1)\ atau\ O(b)$$

dengan n adalah besarnya nilai pangkat (eksponen), dan b adalah panjang digit biner nilai pangkat (eksponen).