MENGIDENTIFIKASI BILANGAN PRIMA-SEMU (PESUDOPRIME) DALAM PENGUJIAN PRIMALITAS MENURUT TEOREMA KECIL FERMAT MENGGUNAKAN *MATHEMATICA*

Ega Gradini[1]

Abstrak

Uji primalitas adalah proses untuk menguji apakah bilangan bulat *n* merupakan bilangan prima atau komposit. Beberapa uji primalitas seperti tes Fermat, Miller-Rabin, dan Lucas-Lehmer yang merupakan uji primalitas probabilistik memberikan hasil yang relatif lebih tepat dan cepat daripada uji deterministik, berpeluang memberikan bilangan prima palsu/semu, dikenal dengan *pseudoprime*. Untuk melaksanakanpengujian, peneliti merumuskan algoritma Teorema Kecil Fermat lalu algoritma dikodekan dalam *Mathematica* (versi 8.0). Penerapan algoritma Teorema Little Fermat dan Teorema Euler tersebut mengarah pada konsep prima semu (*pseudoprime*). Dengan menggunakan perangkat lunak Mathematica 8.0, ditemukan banyaknya bilangan prima ≤10.000 adalah 1229. Kemudian dengan membagi *n (psp)* pada 1229, persentase pseudo-prime pada setiap basis dikumpulkan. Pada bilangan prima ≤10.000, $2 \leq a \leq 20$, basis 8 menghasilkan jumlah *pseudoprime* tertinggi, yaitu mencapai 5,70%, sedangkan jumlah *pseudoprime* terendah (1,22%) dihasilkan oleh basis 7. Meski prima-semu tidak terlalu banyak, tapi tidak cukup langka untuk diabaikan, mengingat prima-semu merupakan salah satu implikasi Teorema Kecil Fermat.

Kata Kunci :*Uji primalitas, Fermat, Pseudoprime, Mathematica, Bilangan Prima*

Abstract

Primalitas is a test process to test whether an integer n is a prime number or a composite. Some test primalitas like Fermat's test, Miller-Rabin, and Lucas-Lehmer test for primalitas which is a probabilistic relative outcomes more accurately and quickly than deterministic tests, could give a false prime number/pseudo, known with a pseudoprime. For melaksanakanpengujian, the researchers formulate Fermat's Little Theorem then algorithm the algorithm coded in Mathematica (version 8.0). Application of algorithm of Fermat's Little Theorem and Euler's Theorem leads to the concept of prima pseudo (pseudoprime). By using the software Mathematica 8.0, found large number of primes ≤ 10,000 is 1229. Then by dividing n (psp) at 1229. the percentage of pseudo-prime on each base of gathered. On the primes ≤ 10,000, $2 \leq a \leq 20$, base 8 generate the highest amount of pseudoprime, namely achieving 5.70%, while the number of lowest pseudoprime (1.22%) generated by the base 7. Though prima pseudo-not too much, but it's not quite rare to be ignored, given the prima-pseudo is one of Fermat's Little Theorem implications.

***Keywords**: Test Primalitas, 1994, Pseudoprime, Mathematica, Prime Numbers*

[1]Ega Gradini, STAIN Gajah Putih Takengon. Email: ega.gradini@gmail.com

PENDAHULUAN

Persoalan keamanan adalah permasalahan paling penting dalam kehidupan manusia. Hampir semua kegiatan melibatkan masalah keamanan seperti *Personal Identification Number* (PIN) ATM, *Password*, surat elektronik, transaksi kartu kredit, transfer dana, bahkan perintah untuk berperang dengan tentara negara lain. Proses tersebut memerlukan teknologi pensandian yang dikenal dengan Kriptografi. Bilangan Prima memainkan peran penting dalam kriptografi mengingat hingga saat ini, tidak ada formula yang valid untuk menghasilkan bilangan prima. Berbagai metode penentuan bilangan prima telah diajukan, diantaranya dengan uji primalitas atau komposit sebuah bilangan bulat yang diberikan.

Uji primalitas adalah proses untuk menguji apakah bilangan bulat*n*merupakan bilangan prima atau tidak. Baru-baru ini, uji primalitas adalah salah satu masalah penting dalam konsep bilangan prima dan menjadi lebih penting karena aplikasi bilangan prima di beberapa area sehingga seperti mengenkripsi basis data, pemrograman komputer, mengkonstruksi perangkat keras dan perangkat lunak, mendeteksi kesalahan dalam pengkodean, kunci keamanan dalam kriptografi, dan keamanan informasi.

Terdapat dua jenis uji primalitas yaitu uji deterministik dan probabilistik. Uji deterministik adalah uji primalitas yang menentukan dengan pasti apakah sebuah bilangan bulat adalah bilangan prima atau tidak. Uji Lucas-Lehmer adalah salah satu uji deterministik. Uji probabilistik juga menentukan apakah bilangan *n* adalah bilangan prima atau tidak, namun uji probabilistik berpotensi (walaupun dengan probabilitas sangat kecil) salah mengidentifikasi bilangan komposit sebagai prima (tidak berlaku sebaliknya). Namun, pada umumnya uji probablistik jauh lebih cepat daripada uji deterministik. Uji Fermat, Uji Solovay-Strassen, dan Uji Miller-Rabin adalah beberapa contoh uji probabilistik.

Pierre de Fermat (1601-1665) adalah salah satu matematikawan paling terkenal di dunia karena karya-karyanya pada teori bilangan, aljabar, kalkulus, probabilitas dan geometri analitik. Karya yang paling terkenal adalah Fermat Little Theorem dan Fermat Last Theorem yang menjadi salah satu teori fundamental dalam pengujian primalitas.

Uji Fermat adalah uji probabilistik karena tes ini tidak dapat secara pasti mengidentifikasi bilangan yang diberikan merupakan prima, terkadang gagal. Hal ini disebabkan teorema Little Fermat dan teorema Euler tidak berlaku dua arah. (Jones, 1998: 126)

KAJIAN PUSTAKA

1. Teorema Banyaknya Bilangan Prima

Misalkan $\pi(x)$ menyatakan banyaknya bilangan prima $\leq x$, maka untuk nilai x yang besar, $\pi(x)$ dihitung dengan $x \ln x$ (Bektas, 2005: 90). Dengan menggunakan sintaks built-in yang terdapat dalam *Mathematica* 8.0, diperoleh banyaknya bilangan prima $\leq x$, dinyatakan dengan *n(x)*. Pada tabel 1 berikut, disajikan perbandingan banyaknya bilangan

prima ≤ *x* antaramenggunakan teorema 1.1 dan *Mathematica* 8.0.

Tabel 1. Banyaknya bilangan prima $\pi(x)$, $x \leq 10{,}000{,}000{,}000$ antara teorema 1.1 dan *Mathematica* 8.0

x	*Teorema 1.1*	*Mathematica 8.0*	*Selisih*
10	4	4	0
100	22	25	3
1,000	145	168	23
10,000	1,086	1229	143
100,000	8,686	9592	906
1,000,000	72,383	78,498	6,115
10,000,000	620,421	664,579	44,158
100,000,000	5,428,682	5,761,455	332,773
1,000,000,000	48,254,943	50,847,534	2,592,591
10,000,000,000	434,294,482	455,052,511	20,758,029

Tabel 1 menunjukkan perbedaan banyaknya bilangan prima yang cukup signifikan antara teorema 1.1 dan *Mathematica* 8.0. Dalam penelitian ini, banyaknya bilangan prima yang menjadi acuan dalam mengidentifikasi prima semu adalah yang dihasilkan oleh *Mathematica* 8.0

2. TeoremaTeorema Kecil Fermat (*Fermat's Little Theorem*)

Jika *p* adalah bilangan prima dan *a* adalah bilangan bulat, maka $a^p \equiv a \pmod{p}$. Selanjutnya, jika pembagi bersama terbesar dari *a* dan *p* adalah 1, maka $a^{(p-1)} \equiv 1 \pmod{p}$ (Mcintosh, 2007).

Bukti teorema dapat ditemukan dalam buku aljabar apa pun, berikut ini salah satunya.

Bukti :

Jika *p*-1 perkalian positif dari a,maka bilangan terdapat bilangan bulat

$$a, 2a, 3a, \ldots, (p-1)a.$$

Jika *ra* dan *sa* keduanya modulo *p*, maka

$$ra = sa \pmod{p},\ 1 \leq r < s \leq p-1$$

dengan habis membagi kedua ruas dengan *a* dan menghasilkan $r \equiv s \pmod{p}$, dimana ini tidak mungkin terjadi. Oleh karena itu, himpunan bilangan bulat sebelumnya haruslah kongruen modulo p terhadap 1, 2, 3, …,$p-1$ dengan susunan yang sama. Dengan mengalikan bagian-bagian ini, diperoleh

$$a.2a.3a\ldots(p-1)\,a \equiv 1.2.3\ldots.(p-1) \pmod{p}$$

maka

$$a^{(p-1)}(p-1)! \equiv (p-1)! \pmod{p}$$

dengan membagi kedua ruas dengan (*p*-1)!,

karena $p \nmid (p-1)!$ diperoleh :

$$a^{(p-1)} \equiv 1 \pmod{p}$$

(Dorsey, 1999).

Teorema Fermat memungkinkan pembuktian bahwa bilangan *n* tertentu adalah komposit tanpa memfaktorkannya. Teorema Kecil Fermat dapat diubah dalam pernyataan alternatif, jika $a^{(n-1)} \not\equiv 1 \pmod{n}$ untuk beberapa *a* dimana $a \not\equiv 0 \pmod{n}$ maka *n* adalah komposit.

Teorema Kecil Fermat mengatakan bahwa jika *n* adalah bilangan prima maka $a^n \equiv a \pmod{n}$. Teorema Euler juga mengatakan bahwa jika *p* adalah bilangan prima dan *a* adalah bilangan bulat, maka $a^p \equiv a \pmod{p}$. Selanjutnya, jika pembagi bersama terbesar

dari *a* dan *p* adalah 1, maka $a^{(p-1)} \equiv 1 \pmod{p}$ (Mcintosh, 2007 : 34).

Teorema ini tidak menjamin primalitas *n* bahkan jika *n* memenuhi kongruensi. Oleh karena itu, teorema ini tidak berlaku dua arah, namun pengujian tersebut mengasumsikan berlaku. Hal ini mengakibatkan munculnya prima-semu (*pseudoprime*).

METODE PENELITIAN

Penelitian ini merupakan penelitian pengembangan (*Research and Development*) yang bertujuan untuk merancang algoritma dan kode tes Fermat yang berbasis Teorema Kecil Fermat *(Fermat's Little Theorem*).Algoritma dan kode yang dirancang lalu digunakan untuk mengidentifikasi *pseudoprime* (prima semu) dalam barisan bilangan yang dinyatakan "Prima" oleh Teorema Kecil Fermat.

Pada penelitian ini, tes Fermat diterapkan dengan bantuan perangkat lunak *Mathematica* 8.0 untuk menilai kemampuannya. *Mathematica* 8.0 sangat mudah dipelajari dan perintahnya sangat sederhana. *Mathematica* 8.0 memiliki begitu banyak sintaks*built-in* yang dapat melakukan banyak tugas teknis seperti menghitung jumlah digit, menyelesaikan masalah kongruensi dan modulo, dimana fungsi ini sangat dibutuhkan dalam melakukan pengujian bilangan prima. Penelitian ini berlangsung dalam beberapa tahap, yaitu:

Pertama, peneliti merancang algoritma dan kode sumber (*source code*) yang digunakan dalam pengujian. Kode sumber dalam penelitian ini berasal dari algoritma yang diperoleh dari teorema Fermat, Teorema Euler dan teorema terkait lainnya dalam teori bilangan.

Kedua, kode sumber/ kode program digunakan untuk menguji apakah bilangan yang diberikan adalah bilangan prima. 100 bilangan bulat pertama ditetapkan sebagai *input*. *Output*nya akan muncul sebagai prima atau komposit. Setelah ini selesai peneliti memperbesar rentang input ke 1000 pertama dan sampai 10.000. Peneliti juga menggunakan beberapa sintaks*built-inMathematica*8.0 untuk melakukan beberapa tugas teknis, seperti menemukan jumlah bilangan prima, menentukan bilangan prima, *pseudoprime*, dan bilangan Carmichael yang diperoleh dari *output* kode sumber. Perintah "PrimeQ [integer]" dan "PrimePi[integer]" digunakan untuk membandingkan daftar bilangan prima dan *pseudoprime*, dan menghitung persentase *pseudoprime* yang dihasilkan oleh setiap pengujian. Untuk mendapatkan gambaran lengkap tentang algoritma dan kode program, peneliti menyarankan agar mengacu pada (Gradini, 2009: 32-65).

Ketiga, menganalisa dan mengidentifikasikan prima semu yang terdapat pada barisan bilangan yang lulus pengujian primalitas.

HASIL PENELITIAN

1. **Uji Fermat dan Koding dalam *Mathematica* 8.0**

Uji Fermat dikembangkan dari Teorema Kecil Fermat dan kemudian menjadi salah satu uji prima probabilistik. Menurut Teorema Kecil Fermat, jika *n* prima dan GCD (*a*, *n*) = 1maka $a^{(n-1)} \equiv 1 \pmod{n}$.

Jika *n* tidak prima, tidak perlu benar bahwa $a^{(n-1)} \equiv 1$, namun masih ada

kemungkinan.Menurut Herman dan Soltys (2008), semua bilangan prima melewati uji Fermat untuk semua$a \in \mathbb{Z}$. Teorema Fermat juga bisa digunakan untuk menguji komposit dari sebuah bilangan. Untuk bilangan bulat tertentu *n*, pilih beberapa bilangan bulat *a* dengan GCD (*a, n*) = 1 dan hitung $r \equiv a^{n-1} (\text{mod } n)$. Jika perhitungan modulo *n* memberikan hasilnya tidak sama dengan 1, *n* adalah komposit. Jika tidak, *n* mungkin prima, dengan kata lain, *n* bisa prima atau komposit.

Berdasarkan langkah-langkah diatas, peneliti menyusun algoritma Uji Fermat sebagai beikut.

Input: an integer n ≥ 3.

Output: n is prime or n is composite

1) *Choose random integer a with 2 ≤ a ≤ n – 1 and GCD (a, n) = 1.*
    a. *Compute* $r \equiv a^{n-1} \text{ (mod n)}$.
    b. *If r ≠ 1, n is composite, otherwise n is prime number.*
2) *If GCD (a, n) ≠ 1, then n is composite.*

Disini, ketika GCD(*a, n*) ≠ 1, *n* adalah komposit karena *n* memiliki pembagi lain selain 1 dan dirinya sendiri, yang berkontrakdiksi dengan definisi bilangan prima. Jika GCD (*a, n*) = 1, *n* bisa menjadi bilangan prima atau komposit. Peneliti lalu mengubah algoritma uji Fermat diatas menjadi kode program dalam Mathematica 8.0. Berikut ini adalah kode sumber (*source code*) uji Fermat.

```
a=__;
n=__;
If [2≤a≤n-1,
If [GCD [a,n]==1,
r =PowerMod [a,n-1,n];
If[r≠1,n "is composite",
n "is Prime"],
n "is composite "],
"cannot be proceed, pick a any integer 2≤a≤n-1"]
```

Ketika kode program ini dijalankan, jika input adalah bilangan prima maka akan memberitahu bahwa *n* adalah prima.Jika tidak maka akan memberi pesan *n* adalah komposit. Ternyata, ada bilangan yang juga diidentifikasi sebagai bilangan prima padahal bukan. Bilangan ini disebut *pseudoprimes*. Hal ini disebabkan lagi oleh teorema Fermat.

Kode sumber ini juga bisa mengidentifikasi Bilangan Carmichael (prima-semu absolut). Bilangan Charmichael adalah pseudoprime untuk semua basis *a*. Dari keluaran yang dihasilkan oleh pengujian ini, kita memiliki daftar bilangan prima untuk setiap basis *a* yang kurang dari 10.000. Perintah *build-inMathematica*, "PrimeQ [integer]" digunakan untuk memverifikasi primality hasil tes dan dari sini dapat diidentifikasi prima semu, maka perintahnya, "PrimePi[10.000]", digunakan untuk menghitung persentase *pseudoprime*. Semakin kecil presentasinya, semakin baik kemampuan tesnya, karena ini mengindikasikan keakuratan tes. Untuk mengidentifikasi bilangan Carmichael, hanya perlu mencari tahu prima semu bersama untuk setiap basis *a*. Di sini, penulis membatasi pilihan basis *a* dari 2 sampai 20 saja. Untuk daftar keluaran lengkap *pseudoprime* dan bilangan Carmichael lihat (Gradini, 2009).

2. Bilangan Prima Semu dalam pengujian primalitas

Jika *n* adalah bilangan prima semu Fermat (disederhanakan menjadi prima semu) maka $a^{n-1} \equiv 1 (\text{mod } n)$ maka *n*adalah bilangan komposit. Secara umum, bilangan komposit *n* yang memenuhi $a^{n-1} \equiv 1(\text{mod } n)$ adalah bilangan prima semu pada basis *a*. Prima semu muncul karena Teorema Kecil Fermat berlaku satu arah, mengingat teorema menyatakan jika *n* bilangan prima maka$a^{n-1} \equiv 1(\text{mod } n)$, tetapi tidak harus benar jika $a^{n-1} \equiv 1(\text{mod } n)$, *n* merupakan bilangan prima.

Contoh 1.

1. 341 adalah prima semu pada basis 3. Tentu saja 341 adalah komposit karena 341 = 11× 31,tetapi 341 lulus uji Fermat karena $3^{341-1} \equiv 1(\text{mod } 341)$.
2. 217 adalah prima semu pada basis 5. 217 = 7× 31, maka 217 adalah komposit, tetapi $5^{217-1} \equiv 1(\text{mod } 217)$,maka217 memenuhi uji Fermat.

Peneliti lalu menjalan kode program hingga basis 20 untuk bilangan n<10,000 sehingga diperoleh bilangan prima semu pada setiap basis sebagaimana disajikan pada tabel berikut.

Tabel 2. Hasil identifikasi prima semu <10,000 pada $2 \leq a \leq 20$

a	Pseudo-prime ≤ 10 000 (Psp)	n (Psp)	%
2	341, 561, 645, 1105, 1387, 1729, 1905, 2047, 2465, 2701, 2821, 3277, 4033, 4369, 4371, 4681, 5461, 6601, 7957, 8321, 8481, 8911	22	1.79%
3	91, 121, 286, 671, 703, 949, 1105, 1541, 1729, 1891, 2465, 2665, 2701, 2821, 3281, 3367,3751, 4961, 5551, 6601, 7381, 8401, 8911	23	1.87%
4	15, 85, 91, 341, 435, 451, 561, 645, 703, 1105, 1247, 1271, 1387, 1581, 1695, 1729, 1891, 1905, 2047, 2071, 2465, 2701, 2821, 3133, 3277, 3367, 3683, 4033, 4369, 4371, 4681, 4795, 4859, 5461, 5551, 6601, 6643, 7957, 8321, 8481, 8695, 8911, 9061, 9131, 9211, 9605, 9919	47	3.82%
5	124, 217, 561, 781, 1541, 1729, 1891, 2821, 4123, 5461, 5611, 5662, 5731, 6601, 7449, 7813, 8029, 8911, 9881	19	1.55%
6	35, 185, 217, 301, 481, 1105, 1111, 1261, 1333, 1729, 2465, 2701, 2821, 3421, 3565, 3589, 3913, 4123, 4495, 5713, 6533, 6601, 8029, 8365, 8911, 9331, 9881	27	2.20%
7	25, 325, 561, 703, 817, 1105, 1825, 2101, 2353, 2465, 3277, 4525, 4825, 6697, 8321	15	1.22%
8	9, 21, 45, 63, 65, 105, 117, 133, 153, 231, 273, 341, 481, 511, 561, 585, 645, 651, 861, 949, 1001, 1105, 1281, 1365, 1387, 1417, 1541, 1649, 1661, 1729, 1785, 1905, 2047, 2169,2465, 2501, 2701, 2821, 3145, 3171, 3201, 3277, 3605, 3641, 4005, 4033, 4097, 4369,4371, 4641, 4681, 4921, 5461, 5565, 5963, 6305, 6533, 6601, 6951, 7107, 7161, 7957, 8321, 8481, 8911, 9265, 9709, 9773, 9881, 9945	70	5.70%
9	28, 52, 91, 121, 205, 286, 364, 511, 532, 616, 671, 697, 703, 946, 949, 1036, 1105, 1288, 1387,1541, 1729, 1891, 2465, 2501, 2665, 2701, 2806, 2821, 2926, 3052, 3281, 3367, 3751, 4376, 4636, 4961, 5356 ,5551, 6364, 6601, 6643,7081, 7381, 7913, 8401, 8695, 8744, 8866, 8911	49	3.99%
10	33, 91, 99, 259, 451, 481, 561, 657, 703, 909, 1233, 1729, 2409, 2821, 2981, 3333, 3367, 4141, 4187, 4521, 5461, 6533, 6541, 6601, 7107,	30	2.44%

a	**Pseudo-prime ≤ 10 000 (Psp)**	**n (Psp)**	**%**
	7471, 7777, 8149, 8401, 8911		
11	15, 70, 133, 190,259, 305, 481, 645, 703, 793, 1105, 1330, 1729, 2047, 2257, 2465, 2821, 4577, 4921, 5041, 5185, 6601, 7869, 8113, 8170, 8695, 8911, 9730	28	2.28%
12	65, 91, 133, 143, 145, 247, 377, 385, 703, 1045, 1099, 1105, 1649, 1729, 1885, 1891, 2041,2233, 2465, 2701, 2821, 2983, 3367, 3553, 5005, 5365, 5551, 5785, 6061, 6305, 6601, 8911, 9073	33	2.69%
13	21, 85, 105, 231, 244, 276, 357, 427, 561, 1099, 1785, 1891, 2465, 2806, 3605, 5028, 5149, 5185, 5565, 6601, 7107, 8841, 8911, 9577, 9637	25	2.03%
14	15, 39, 65, 195, 481, 561, 781, 793, 841, 985, 1105, 1111, 1541, 1891, 2257, 2465, 2561, 2665, 2743, 3277,5185, 5713, 6501, 6533, 6541, 7107, 7171, 7449, 7543, 7585, 8321, 9073	32	2.60%
15	341, 742, 946, 1477, 1541, 1687, 1729, 1891, 1921, 2821, 3133, 3277, 4187, 6541, 6601, 7471, 8701, 8911, 9073	19	1.55%
16	51, 85, 91, 255, 341, 435, 451, 561, 595, 645,703, 1105, 1247 , 1261, 1271, 1285, 1387, 1581, 1687, 1695, 1729, 1891, 1905, 2047, 2071, 2091, 2431, 2465, 2701, 2821, 3133, 3277, 3367, 3655, 3683, 4033, 4369, 4371, 4681, 4795, 4859, 5083, 5151, 5461, 5551, 6601, 6643, 7471, 7735, 7957 , 8119, 8227, 8245, 8321, 8481, 8695, 8749, 8911, 9061, 9131, 9211, 9605, 9919	63	5.13%
17	45, 91, 145, 261, 781, 1111, 1228, 1305, 1729, 1885, 2149, 2821, 3991, 4005, 4033, 4187, 4912,5365, 5662, 5833, 6601, 6697, 7171, 8481, 8911	25	2.03%
18	25, 49, 65, 85, 133, 221, 323, 325, 343, 425, 451, 637, 931, 1105, 1225, 1369, 1387, 1649, 1729, 1921, 2149, 2465, 2701, 2821, 2825, 2977, 3325, 4165, 4577, 4753, 5525, 5725, 5833, 5941, 6305, 6517, 6601, 7345, 8911, 9061	40	3.25%
19	45, 49, 153, 169, 343,561, 637, 889, 905, 906, 1035, 1105, 1629, 1661, 1849, 1891, 2353, 2465, 2701, 2821, 2955, 3201, 4033, 4681, 5461, 5466, 5713, 6223, 6541, 6601, 6697, 7957, 8145, 8281, 8401, 8869, 9211, 9997	38	3.09%
20	21, 57, 133, 231, 399, 561, 671, 861, 889, 1281, 1653, 1729, 1891, 2059, 2413, 2501, 2761, 2821, 2947, 3059, 3201, 4047, 5271, 5461, 5473, 5713, 5833, 6601, 6817, 7999, 8421, 8911	32	2.60%

Tabel 2 memberikan distribusi prima-semu <10.000 pada setiap basis *a*, banyaknya prima-semu dinyatakan dengan *n(psp)*. Semua bilangan prima-semu tersebut diperoleh dengan membandingkan keluaran Uji Fermat dengan daftar bilangan prima <10.000 yang berjumlah 1229 bilangan untuk setiap basis. Hasil komparasi menunjukkan bahwa pada basis 2, terdapat 22 (1,79%) bilangan prima-semu, pada basis 3 terdapat 23(1,87%) prima-semu dan pada basis 4 jumlah prima-semu relative tinggi yakni 47 (3,82%) dari 1229 bilangan prima pada basis tersebut. Prima-semu terbanyak terdapat pada 8 berjumlah 70 (5.70%) dan pada basis 16 berjumlah 63(5,13%). Meski jumlah prima-semu yang dihasilkan oleh Uji Fermat tidaklah banyak, namun "kesalahan" ini tidak bisa diabaikan.

KESIMPULAN

Dengan menggunakan *Mathematica* 8.0 untuk uji primalitas, peneliti berhasil (1) menentukan banyaknya bilangan prima hingga 10,000,000, (2) merancang dan mengeksekusi program uji Fermat, dan (3)mengidentifikasi prima-semu pada keluaran bilangan prima yang dihasilkan oleh uji Fermat. Tugas teknis ini biasanya dilakukan dengan menggunakan bahasa pemrograman yang memakan waktu lama untuk melakukan so*urce cod*e. Misalnya bahasa pemrograman C atau C ++ dan bahasa pemrograman lainnya yang membutuhkan pemahaman mendalam dan memakan waktu. Dari hasil penelitian diperoleh bahwa pada bilangan prima ≤ 10.000, $2 \leq a \leq 20$, basis 8 menghasilkan jumlah pseudoprim terbesar, yaitu mencapai 5,70%, sedangkan jumlah pseudoprim terkecil (1,22%) yang dihasilkan oleh basis 7.

DAFTAR PUSTAKA

Bektas,Attila. 2005. *Probabilistic Primality Test*. Master's Thesis. The Middle East Technical University.

Eynden,Charles Vanden. 2001. *Elementary number theory.2nd edn.*Illnois : McGraw-Hill

Gradini,Ega. 2009. *Primality Testing.* Project report of MSc in Teaching of Mathematics, Universiti Sains Malaysia.

Jones,Gareth A &Jones, Mary J. 1998. *Elementary Number Theory*.London:springer-Verlag

Kumanduri,Ramanujan&Romero,cristina. 1998. *Number Theory with Computer Application*.New Jersey : Prentice-Hall

Lenstra, H.W.Jr. 1997. *Primality Testing*. [Accessed 5th January 2017], diperoleh melalui World Wide Web :https://openaccess.leidenuniv.nl/space/bitstream/1887/3818/1/346_071.pdf

McIntosh,Christina. 2007. *finding Prime Numbers: Miller-Rabin and Beyond*.Electronic Journal of Undergraduate Mathematics.[Online],2007(12).[diakses pada 20 Oktober 2017], diperoleh melalui World Wide Web : www.scribd.com/doc/4904376/FINDING-PRIME-NUMBER -