JEMS (Jurnal Edukasi Matematika dan Sains), 9(2), 2021, 499-506
DOI: 10.25273/jems.v9i2.11449

Aplikasi Modulo Berpangkat a^b mod n Menggunakan Pola Barisan dan Teorema Euler Berbasis Web

Rohmad Wahid Rhomdani, Yoga Dwi Windy Kusuma Ningtyas

Abstrak:
Penelitian mengunakan adalah studi literatur yaitu mengkaji teorema Euler dan menguji serta menyelidiki pola barisan pada modulo berpangkat a^b mod n. Peneliti melakukan pengujian dengan javscript untuk menentukan n dan a adalah bilangan bulat positif yang saling koprima dan dinyatakan sebagai $a^{\varphi(n)} \equiv 1 \pmod{n}$ atau $a^{\varphi(n)}$ mod $n = 1$. Peneliti berhasil merangkai algoritma javascript dan membuat aplikasi berbasis web dapat memberikan jawaban yang akurat pada penyelesaiannya modulo berpangkat dengan dua cara yaitu sisa pembagian pangkat menggunakan pola barisan dan menggunakan cara penyelesaian pangkat fungsi phi Euler $\varphi(n)$ modulo n yang kongruen dengan satu.

Kata Kunci : Modulo, Barisan, Euler, Javascript dan Web

Abstract:
The research used is a literature study, which examines Euler's theorem and examines and investigates the sequence pattern on the modulo to the power of a^b mod n. The researcher conducted a test using JavaScript to determine that n and a are positive integers that are mutually compatible and expressed as $a^{\varphi(n)} \equiv 1 \pmod{n}$ or $a^{\varphi(n)}$ mod $n = 1$. The researcher succeeded in compiling a javascript algorithm and creating a web-based application. can give an accurate answer on the solution modulo to the power of the two ways, namely the remainder of the division of the power using a sequence pattern and using the method of solving the power function of phi Euler $\varphi(n)$ modulo n which is congruent to one.

Keywords : *Modulo, line, Euler, Javascript and Web*

Pendahuluan

Perkembangan teknologi yang begitu pesat tidak lepas dari pengaruh matematika begitu juga diera society 5.0 perkembangan teknologi turut mendorong pembelajaran matematika karena itu untuk mengembangkan dan menciptakan teknologi dijaman sekarang ini diperlukan penguasaan matematika yang begitu kuat sejak dini. Menilai betapa pentingnya peran matematika dalam mengikuti perkembangan teknologi, maka kita harus juga ikut andil dalam menerapkan aplikasi khususnya javascript yang begitu populer digunakan oleh para programmer web dapat menerapkannya aplikasi javascript pada pembelajaran matematika (Himawan, 2019)

Penerapan aplikasi javascript pada pembelajaran matematika karena bahasa pemrogramannya sangat mudah digunakan, dan bisa juga digunakan untuk mengembangkan aplikasi matematika berbasis website hampir semua aplikasi yang digunakan membuat atau membangun sebuah website dibuat menggunakan Javascript. (Mariko S, 2019).

Javascript didukung oleh hampir semua browser, sehingga javascript menjadi bahasa pemrograman yang universal, ini membuat logika bahasa pemrograman javascript lebih di-

Rohmad Wahid Rhomdani, Universitas Muhammadiyah Jember
wahidgrup@gmail.com

Yoga Dwi Windy Kusuma Ningtyas, Universitas Muhammadiyah Jember
kusumaningtyas.dwi@unmuhjember.ac.id

r
l

yang lebih powerful. karena toolsnya yang universal. Javascript gratis untuk digunakan dan semua resource-nya dapat kita pelajari tanpa harus mengikuti pelatihan. sehingga Javascript lebih ungggul dan populer digunakan para programmer web, dan sangat cocok digunakan untuk mengembangkan aplikasi matematika berbasis website (Padmanaba A, 2020).

Salah satu inovasi dalam mengembangkan aplikasi pembelajaran matematika yang cocok diera scociety 5.0 yaitu mengembangkan aplikasi modulo berpangkat a^b mod n menggunakan javascript berbasis website. Peneliti sangat tertarik untuk mengembangan hal baru yaitu menggunakan pola barisan untuk bisa memecahkan modulo berkapat a^b mod n. Sehingga peneliti merumuskan masalah bagaimana tahapan membangun aplikasi menggunakan javascript dan bagaimana hasil uji coba aplikasi modulo berkapat pada a^b mod n menggunakan pola barisan dan teorema Euler berbasis website.

Metode

Metode penelitian yang digunakan adalah studi literatur yaitu mengkaji teorema yang ada dan menguji dengan materi dari jurnal atau buku-buku teks yang berkaitan dengan pembahasan ini untuk menyelidiki adakah pola barisan pada modulo berpangkat a^b mod n dan apakah sesuai dengan Teorema Euler. Fungsi Euler Totient atau Euler Phi diperkenalkan oleh Leonhard Euler yang menyatakan bahwa jika n dan a adalah bilangan bulat positif yang saling koprima, maka a pangkat fungsi phi Euler dari n akan kongruen dengan satu dalam modulo n. Hal ini dapat dinyatakan sebagai $a^{\varphi(n)} \equiv 1 \pmod{n}$ atau $a^{\varphi(n)}$ mod n = 1. Kemudian dikatakan koprima jika (a,n) bilangan bulat atau relatif prima atau saling prima dan memiliki nilai FPB kedua bilangan tersebut adalah 1. (Chung F & Folkman J, 2018)

Hasil dan Pembahasan

Hasil dan pemababahasan pada penelitian ini yaitu menjabarkan tahapan-tahapan pembuatan algoritma pada javascript untuk menampilkan soal serta bagaimana cara penyelesaiannya menggunakan

```
var a=7;var b=5000;var n=13;
document.write(a+'<sup>'+b+'</sup>'+' mod '+n);
```

Algoritma *Javascript* diatas adalah cara menampilkan soal modulo dengan input data nilai: a=7; b=5000; n=13; sehingga jika dirunning akan tampil seperti berikut : 7^{5000} mod 13 ini adalah cara menampilkan soal modulo pada *Javascript*. Kemudia bagaimana mencari solusinya 7^{5000} mod 13 dengan javascript, berikut peneliti memaparkan tahap demi tahap cara penyelesaiannya untuk 7^{5000} mod 13 sebelum itu peneliti menguji pangkat 5000 mengujinya dengan menggunakan looping pada javascript pada nilai b : 0<b<19. berikut ini adalah algortima *Javaccript* 7^{19} mod 13 :

```
var a=7; var n=13; let text='';for(b=0;b<18;b++){var c=a**b;
var d=c%n;text+=a+'<sup>'+b+'</sup>&rarr;'+c+'mod'+n+'='+d+'<br>';}
document.write(text);
```

Algoritma diatas cara menampilkan a^b mod $n \equiv 7^{18}$ mod 13 . Hal ini digunakan untuk menguji dan menentukan banyaknya perulangan pada 7^{18} mod 13 dengan nilai b : 0<b<18 Kemudain kita menggunakan loop pada javascript, dari keempat loop (*For,While, Do While & forEach*) peneliti menggunakan looping *For* karena peneliti membutuhkan pengujian data dengan batasan nilai b : 0<b<18. Nilai pada [`var c=a**b;`] script ini untuk menghitung nilai a pangkat b atau (a^b), sedangkan nilai d pada [`var d=c%n;`] script ini artinya untuk menghitung hasil dari nilai [*c* mod *n*]. Kemudian baru kita loop mulai nilai *b* : 0<*b*<18. keterangan dari hasil yang akan di running menggunakan *loop* pada *Javascript* berikut ini.

```
[a+'<sup>'+b+'</sup>&rarr;'+c+'mod'+n+'='+d+'<br>';]
```

7^{18} → 5764801 mod 13 = 3

pada saat pengujian menggunakan loop `for(b=0;b<18;b++)]` peneliti berhasil mendapatkan jawaban yang diharapkan, proses pengujian ditentukan dengan input nilai : a^b mod n → 7^{18} mod 13 berikut adalah contoh pada aplikasi (a^b mod n) dengan angka yang sudah ditentukan a = 7; b = 18; n = 13 algoritma ini digunakan untuk menguji angka berulang pada 7^{18} mod 13 pada *Javascript*, sehingga setelah di running jawaban dari 7^{18} mod 13 terdapat angka berulang dan memiliki pola barisan yaitu sebagai berikut :

Tabel 1. Hasil pengujian 7^b mod 13.

no	a^b mod n → 7^{18} mod 13	Perulangan
1.	7^0 → 1 mod 13 = 1	1
2.	7^1 → 7 mod 13 = 7	7
3.	7^2 → 49 mod 13 = 10	10
4.	7^3 → 343 mod 13 = 5	5
5.	7^4 → 2401 mod 13 = 9	9
6.	7^5 → 16807 mod 13 = 11	11
7.	7^6 → 117649 mod 13 = 12	12
8.	7^7 → 823543 mod 13 = 6	6
9.	7^8 → 5764801 mod 13 = 3	3
10.	7^9 → 40353607 mod 13 = 8	8
11.	7^{10} → 282475249 mod 13 = 4	4
12.	7^{11} → 1977326743 mod 13 = 2	2
13.	7^{12} → 13841287201 mod 13 = 1	1
14.	7^{13} → 96889010407 mod 13 = 7	7
15.	7^{14} → 678223072849 mod 13 = 10	10
16.	7^{15} → 4747561509943 mod 13 = 5	5
17.	7^{16} → 33232930569601 mod 13 = 9	9
18.	7^{17} → 232630513987207 mod 13 = 11	11
	Total Perulangan	18

Dari hasil uji coba diatas pada a^b mod n → 7^{18}mod 13 terdapat angka berulang dan memiliki pola barisan jumlah angka tersebut memiliki panjang 18 angka. Namun dari hasil uji coba ini terdapat temuan yaitu aplikasi ini memiliki kekurangan dalam menghitung nilai modulo seperti contoh : 7^{25} → 1.341068619663965e+21 mod 13 = 3 → tidak akurat. Hasil uji coba pangkat maximal menggunkan Javascript nilai modulo berpangkat eksponen seperti contoh 1.341068619663965e+21 belum memberikan jawaban yang akurat dan bisa dilanjutkan penelitian selanjutnya. Kemudian temuan lain untuk mendapatkan hasil yang akurat pada mengujian a^b mod n hasil dari modulo berpangkat tersebut harus memiliki pola barisan yang sama setelahnya, seperti contoh yang telah ditemukan pada soal : 7^{18} mod 13. Pada algoritma ini juga bisa menguji nilai b lebih dari angka 18 namun peneliti hanya menguji dengan angka b<18 karena pengujian modulo a^b mod n → 7^{18} mod 13 sudah cukup dan memenuhi untuk mendapatkan angka yang berulang dan memiliki pola barisan. Jumlah angka pada hasil uji coba pada modulo a^b mod n → 7^{18} mod 13 terdapat 18 angka yang berulang dan ada yang duplikat (sama), seperti pengujian tabel diatas b<18 terdapat 18 angka yang muncul, uji coba tersebut terdapat angka yang sama dan berulang/terduplikat sehingga peneliti melakukan filter data atau menyaring kembali dari 18 angka tersebut diambil angka yang tidak sama atau tidak berulang. Peneliti mendapatkan 18 angka yang terduplikat dengan bantuan algoritma *Javascript*. Berikut hasil perulangan modulo a^b mod n $\rightarrow$ 7^{18} mod 13 = 18 angka, berikut ini :

$$7^{18} \bmod 13 = \underbrace{(1,7,10,5,9,11,12,6,3,8,4,2,1,7,10,5,9,11)}_{\text{ada 18 angka yang berulang (terduplikat).}}$$

karena dari 18 angka tersebut ada yang sama (terduplikat), maka peneliti melakukan filter atau penyaringan dari 18 angka tersebut dengan menghitung nilai peristiwa atau banyaknya angka yang muncul menggunakan algoritma nilai frekuensi pada *Javascript* sebagai berikut :

```
var arr =[1,7,10,5,9,11,12,6,3,8,4,2,1,7,10,5,9,11];
function foo(arr) {var freq1 = [], freq2 = [], prev;
    arr.sort();
    for ( var i = 0; i < arr.length; i++ ) {
        if ( arr[i] !== prev ) { freq1.push(arr[i]);   freq2.push(1);
        }prev = arr[i];} return [freq1, freq2]; }
var result = foo(arr);var ulang = result[1].length;
document.write('Banyak perulangan pada '+a+ '<sup>b</sup> Modulo '+
n+' = (' + result[0] + ')'+' Berulang = '+ulang+' kali. ');
```

hasil tampilan dari algoritma Javascript diatas : Terdapat angka berulang pada 7^b mod 13 = (1,10,11,12,2,3,4,5,6,7,8,9) Berulang = 12 kali. Peneliti berhasil menghapus angka yang sama (terduplikat) dari 18 angka menjadi 12 angka dan berhasil didapatkan dengan bantuan algoritma *Javascript*. Hasil dari uji coba menggunakan javascript untuk soal 7^{18}mod 13 dengan nilai b : $0<b<18$ menggunakan lopp for(b=0;b<18;b++)pada soal 7^{18}mod 13 nilainya sebagai berikut :

$$7^{18} \bmod 13 = \underbrace{(1,7,10,5,9,11,12,6,3,8,4,2)}_{\text{12 kali perulangan}}$$

Setelah uji coba dilakukan peneliti berhasil mendapatkan jumlah angka yang berulang dari banyaknya perulangan pada modulo 7^b mod 13 = (1,7,10,5,9,11,12,6,3,8,4,2) atau berulangsebanyak 12 kali. Karena modulo 7^b mod 13 berulang 12 kali maka peneliti bisa melakukan uji coba lanjutan pada soal pertama yaitu menguji soal 7^{5000} mod 13 dengan cara menurunkan pangkat 5000 mod 12 = 8 karena sisa 8 maka 7^{5000} mod 13 ≡ 7^8 mod 13 berikut ini peneliti membuat kembali algoritma *Javascript* untuk pengujian :
7^{5000} mod 13 ≡ 7^8 mod 13 dibawah ini :

```
var a=7;var b=5000;var n=13;var w=12
var q=b%w; var l=a**q;var f=l%n;var j=a**b;var k=j%n;
document.write(a+'<sup>b</sup> mod '+n+' &rarr; terdapat '+w+' kali
perulangan, sehingga<br>'+a+'<sup>'+b+'</sup> mod '+n+' &rarr; '+' pangkat
'+b+' mod '+w+' = '+q+',sisa '+q+'<br>'+a+'<sup>'+b+'</sup> mod '+n+' ≡ '+a+
'<sup>' +q+'</sup> mod '+n+'<br>'+a+' <sup>'+b+' </sup> ≡ '+a+
'<sup>'+q+'</sup> mod +n+'<br>' +a+'<sup>'+b+'</sup> ≡ '+l+' mod '+n+' =
'+f+ ' hasilnya adalah '+f );
```

Sehingga dari algoritma diatas peneliti mendapatkan hasil yang akurat untuk 7^{5000} mod 13 kongruen dengan 7^8 mod 13 karena pangkat 5000 mod 12 = 8 sisa 8, angka 12 didapatkan dari banyaknya perulangan 7^b mod 13 berikut setelah di running pada javascript diapatkan hasil sebagai berikut :

7^b mod 13 → terdapat 12 kali perulangan, sehingga
7^{5000} mod 13 → pangkat 5000 mod 12 = 8 , sisa 8
7^{5000} mod 13 ≡ 7^8 mod 13
7^{5000} ≡ 7^8 mod 13
7^{5000} ≡ 5764801 mod 13 = 3 hasilnya adalah 3

Dari hasil uji coba aplikasi algoritma nilai frekuensi modulo berpangkat menggunakan Javascript di atas pada soal 7^{5000} mod 13 = 3.
Kita akan uji kembali dengan metode lain yaitu menggunakan dalil Euler : $a^{\varphi(m)} \equiv 1 \pmod{n}$ metode teorema Euler (juga dikenal sebagai Teorema Fermat-Euler) yaitu *n* dan *a* adalah bilangan bulat positif yang saling koprima, maka $a^{\varphi(m)}$ dari *n* akan kongruen dengan satu dalam modulo *n,* dalil Euler pada soal 7^{5000} mod 13.
Berikut penyelesaiannya menggunakan Javascript:

```
var a=7;b=5000;n=13;var penguji=18
for(b=0;b<penguji;b++){var c=a**b; var d=c%n;if(d==1){var Euler_m1=b;}} var
Euler_a=a;var Euler_m=n;if(Euler_a<Euler_m){ss=Euler_m}else{ss=Euler_a}
let Euler_pbb='';let uji='';var Euler_m2=a%Euler_m1;var syaratEuler = [];
for(j=1;j<=ss;j++){if(Euler_a%j==0 && Euler_m%j==0){syaratEuler.push(j);
Euler_pbb +='Nilai PBB dari '+a+','+n+' : ' + j + ' karena : '+j+' sehingga
φ('+n+') = '+Euler_m1+'<br>';
if (syaratEuler==1){document.write(Euler_pbb);}}}
if (syaratEuler==1){var r0=b;var r1=Euler_m1;
var q1=Math.floor(r0/r1);var r2=r0%r1;var q2=Math.floor(r1/r2);var
hs1=(a**r1)%n; var hs2=(a**r2)%n; document.write(
r0+" = "+q1+" &times; "+r1+" + "+r2+"<br>"+a+'<sup>'+b+'</sup>'+' ≡ ('+
a+'<sup>'+r1+'</sup>)<sup>'+q1+'</sup>&times;'+a+'<sup>'+r2+'</sup>mod
'+n+'<br>'+ a+'<sup>'+b+'</sup>'+' ≡ ('+ a+'<sup>'+r1+'</sup>)<sup>'+
q1+'</sup>&times;'+a+'<sup>'+r2+'</sup>mod'+n+'&rarr;['+a+'<sup>'+r1+'</sup>
mod '+n+' = '+hs1+']<br>'+a+'<sup>'+b+'</sup>'+' ≡ ('+ hs1+')<sup>' +q1+
'</sup> &times;'+a+'<sup>'+r2+'</sup>mod '+n+'<br>'+a+'<sup>'+b+'</sup>'+' ≡
'+a+'<sup>'+r2+'</sup>'+' mod '+n+' = '+hs2+'<br>'+'hasilnya adalah =
'+hs2+'<br><br>');}else{document.write(' Nilai (pbb) dari '+a+','+n+' :
'+syaratEuler+' sehingga tidak memnuhi a<sup>φ(m)</sup> ≡ 1 (mod m) <br>');}
```

Algoritma tersebut menguji nilai PBB dan mencari perulangan hasil akhir pada modulo 7^b Modulo 13 namun peneliti tetap perpedoman pada pengujian perulangan untuk mencari syarat yang diberikan oleh teorema Euler atau dalil Euler : $a^{\varphi(m)} \equiv 1 \pmod{n}$: angka 1 terletak pada suke ke 12.

7^{18} Modulo 13 = $\underbrace{(1,7,10,5,9,11,12,6,3,8,4,2,1,7,10,5,9,11)}_{\text{ada [18] angka}}$

angka [1] terletak pada
pangkat ke [12] → 7^{12} mod13 = 1.

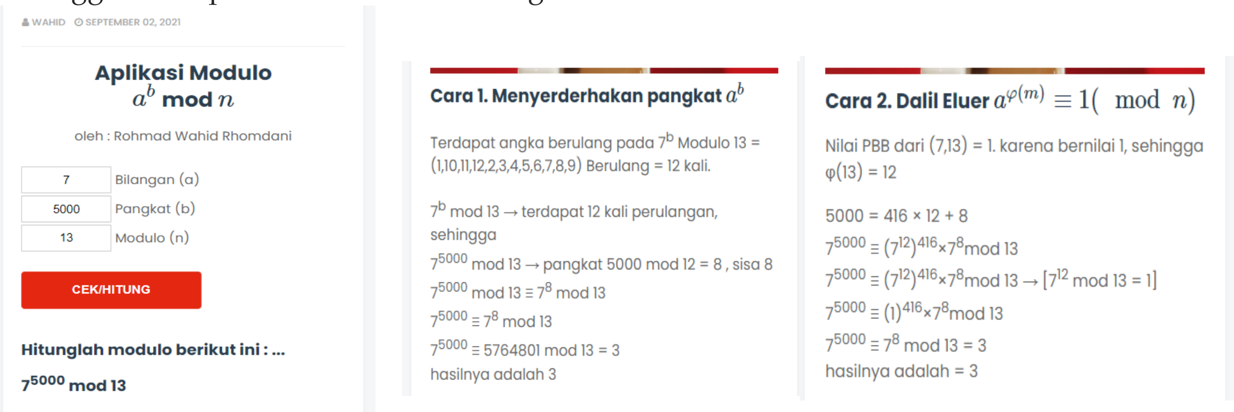
```
var c=a**b; var d=c%n; if(d==1){var Euler_m1=b;}}
```

Fungsi ini mencari syarat yang ditentukan oleh dalil Euler, kemudian peneliti menguji 7^b mod 13 = 1 terletak pada pangkat ke 12. setelah ditemukan letak angka 1 dengan bantuan Javascript di temukanlah *a* pangkat fungsi *phi* Euler atau $7^{\varphi(13)} \equiv 1 \pmod{13}$ terletak pada pangkat ke 12. Sehingga φ(13) = 12.

Kemudian selanjutnya peneliti mencari nilai PBB dari (7,13) = 1 syarat ini adalah syarat mutlak untuk nilai PBB dalil Euler harus sama dengan 1, dengan menggunakan logika pada Javascript mendapatkan nilai PBB dari (7,13) sama dengan 1. Setalah di running algoritma diatas, peneliti dengan mudah mendapatkan hasil yang akurat menggunakan bantuan algoritma Javascript dari soal 7^{5000} mod 13. Jawabannya adalah sebagai berikut :
Nilai PBB dari (7,13) = 1. karena bernilai 1, sehingga φ(13) = 12

$5000 = 416 \times 12 + 8$
$7^{5000} \equiv (7^{12})^{416} \times 7^{8} \bmod 13$
$7^{5000} \equiv (7^{12})^{416} \times 7^{8} \bmod 13 \rightarrow [7^{12} \bmod 13 = 1]$
$7^{5000} \equiv (1)^{416} \times 7^{8} \bmod 13$
$7^{5000} \equiv 7^{8} \bmod 13 = 3$
Hasilnya adalah 3 sama dengan cara penurunan pangkat yaitu hasilnya 3

Aplikasi modulo berpangkat berbasis web pada soal 7^{5000} mod 13 dengan menggunakan pola barisan adalah sebagai berikut :



Gambar 1. Tampilan Apilkasi Moduo Berpangkat dan Penyelesaiannya

Kotak input tersebut bisa juga diisi dengan angka lain secara acak, kemudian untuk menampilkan penyelesaianya tekan tombol cek/hitung pada aplikasi tersebut. Aplikasi akan memberikan dua solusi yang pertama dengan penurunan pangkat yang kedua dengan dalil Euler. Aplikasi ini juga memberikan pengujian deret pada modulo berpangkat a^b mod n sesuai dengan angka yang telah diinputkan. Berikut ini adalah tampilan aplikasi website cara penyelesaian modulo berpangkat dengan menyederhanakan pangkat a^b. Penyelesaian pada aplikasi penyederhanaan pangkat dapat menampilkan data perulangan hasil akhir pada modulo a^b mod n sesuai angka yang telah diinputkan dengan syarat memiliki pola barisan. berikut ini adalah tampilan aplikasi website penyelesaian modulo berpangkat a^b .

Selanjutnya tampilan aplikasi website penyelesaian modulo berpangkat a^b menggunakan dalil Euler. Aplikasi penyelesaian modulo menggunakan dalil Euler ini dapat menampilkan nilai PBB dari dua angka yang telah diinputkan, aplikasi ini juga bisa mencari dan mentukan syarat bilangan bulat positif yang saling koprima atau mencari nilai PBB dari dua angka tersebut yang bernilai 1 seperti contoh PBB dari (7,13) = 1, kemudian aplikasi ini juga bisa mencari dan mentukan syarat dari Euler yaitu pangkat fungsi phi Euler (φ(n)) dari n akan kongruen dengan satu dalam modulo n.

Tabel 2. Perbandingan penurunan pangkat & dalil Euler

Penurunan Pangkat	Dalil Euler
Terdapat angka berulang pada :	Nilai PBB dari (23,31) = 1.
3^b mod 123 = (114,120,27,3,42,81,9,96)	sehingga φ(31) = 12
Berulang = 8 kali.	$9999 = 833 \times 12 + 3$
pangkat 4321 mod 8 = 1	$23^{9999} \equiv (23^{12})^{833} \times 23^{3} \bmod 31$
$3^{4321} \bmod 123 \equiv 3^{1} \bmod 123$	Karena $\rightarrow [23^{12} \bmod 31 = 1]$
$3^{4321} \equiv 3^{1} \bmod 123$	$23^{9999} \equiv (1)^{833} \times 23^{3} \bmod 31$
$3^{4321} \equiv 3 \bmod 123 = 3$	$23^{9999} \equiv 23^{3} \bmod 31 = 15$
hasilnya adalah 3	hasilnya adalah = 15

Terdapat angka berulang pada :	Nilai PBB dari (13,9) = 1.
3^b mod 9 = (1,4,7)	sehingga φ(9) = 12
Berulang = 3 kali.	1313 = 109 × 12 + 5
pangkat 1313 mod 3 = 2	$13^{1313} \equiv (13^{12})^{109} \times 13^5$ mod 9
13^{1313} mod 9 $\equiv 13^2$ mod 9	Karena → [13^{12} mod 9 = 1]
$13^{1313} \equiv 13^2$ mod 9	$13^{1313} \equiv (1)^{109} \times 13^5$ mod 9
$13^{1313} \equiv 169$ mod 9 = 7	$13^{1313} \equiv 13^5$ mod 9 = 7
hasilnya adalah 7	hasilnya adalah = 7

Jawaban dari kedua metode tersebut sama-sama akurat kebenarannya. Berikut ini tampilan website aplikasi Modulo a^b mod n. Dari hasil uji coba aplikasi ini bisa memberikan jawaban yang akurat serta dilengkapi tabel polinomial dari modulo berpangkat a^b mod n. Aplikasi ini juga memberikan penyelesaian modulo berpangkat, sebut pada contoh soal yang telah diinput nilai a=7 b=5000 n=13 pada a^b mod n.

Namun aplikasi ini tidak menggunakan teorema kecil Fermat pada rumus φ(n) = n – 1 untuk n adalah bilangan prima. Justru aplikasi ini mencari nilai phi Euler φ(n) dari mod n yang kongruen dengan nilai 1 atau mencari nilai a^b mod n = 1 dengan pengujian nilai b : 0<b<18 yaitu mencari pangkat berapakah yang nilainya sama dengan satu pada a^b mod n dengan b<18 untuk bisa mendapatkan nilai yang lebih akurat dan banyak cara yang bisa dibuktikan pada penyelesain menggunakan dalil Euler. Sebagai contoh pada soal 7^{5000} mod 13 pada phi Euler φ(n) yaitu phi Euler φ(13) =12 angka 12 tidak menggunakan rumus teorema kecil Fermat yaitu φ(n) = n – 1 namun menggunakan pencarian φ(n) yang bernilai 1 terletak pada pangakat 7^{12} mod 13 = 1. Peneliti menemukan pangkat 12 yang sangat sesuai dan cocok untuk penyelesain dalil Euler, Sehingga rumus phi Euler φ(13) =12. Syarat dari Euler yaitu rumus $a^{\varphi(n)} \equiv 1$ (mod n) yang disebut kongruen dengan satu dalam modulo n menggunakan pengujian a^b mod n dengan nilai b<18.

Simpulan

Kesimpulan dari penelitian ini pada aplikasi modulo berpangkat menggunakan pola barisan dan teorema Euler pada a^b mod n berbasis website adalah sebagai berikut :

1. Aplikasi ini bisa mencari dan menentukan syarat dari Euler yaitu pangkat fungsi phi Euler φ(n) modulo n = 1. Aplikasi ini dapat menentukan syarat bilangan bulat positif yang saling koprima, atau nilai PBB dari (a,n) = 1.
2. Untuk mendapatkan hasil yang akurat pada pengujian a^b mod n hasil dari modulo berpangkat tersebut harus memiliki pola barisan yang sama.
3. Aplikasi ini dapat mencari nilai $a^{\varphi(n)}$ mod n = 1 dari pengujian nilai b : 0<b<18 yaitu pengujian banyaknya angka yang berulang pada modulo a^b mod n agar bisa mendapatkan nilai sama dengan satu yang lebih akurat dan banyak cara yang bisa dibuktikan pada penyelesain menggunakan dalil Euler.
4. Kekurangan dari aplikasi ini jika terdapat jawaban berpangkat eksponen seperti 1.341068619663965e+21 mod n atau diatas ±20 digit, hasilny kurang akurat dan bisa dilanjutkan penelitian selanjutnya.

Daftar Rujukan

Akhil Mathew. 2021. *On K(1)-local TR*. Compositio Mathematica. 157(5)

Chung F, Folkman J & Graham R. 2018. *Sum Sequences Modulo n.* Journal of Combinatorial Theory, Series A. Elsevier. 158(2) : 290-314

Lia Y & Kim D. 2018. *Menon-type identities with additive characters*. Journal of Number Theory.

Elsevier. 192(2):373-385

Mariko S. 2019. *Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus.* Jurnal Inovasi Teknologi Pendidikan. 6(1):80-91

Mustafa S.R. 2020. *Analisis Sisa Pembagian dari Sigma 13i Pangkat m dengan i dari 1 Sampai n oleh Pembagi 2*. Jurnal Ilmiah Edu Research. 9(2):72-77

Padmanaba A, Kumalasari E & Andayati D. 2020. *Komparasi Penggunaan Framework Codeigniter Vs Php Native Pada Sistem Informasi Manajemen Surat Sekretariat Dprd Pemalang*. Jurnal Script. 8(1):1-6

Sumarni A.R. 2020. *Pengembangan Aplikasi Perhitungan Integral Pada Materi Kinematika*. Pendekar: Jurnal Pendidikan Berkarakter. 3(2):58-62