

LAMPIRAN 1

BERITA ACARA BIMBINGAN SKRIPSI



**PANITIA PELAKSANA PROGRAM
FAKULTAS SOSIAL DAN HUMANIORA
UNIVERSITAS NURUL JADID
PROBOLINGGO JAWA TIMUR**

*PP. Nurul Jadid
Keranganyar Paiton
Probolinggo 67291
T. 0886.1077077
sashum@unuja.ac.id*

BERITA ACARA BIMBINGAN SKRIPSI

1. Nama Mahasiswa : **NURUL JANNAH**
2. NIM : **1942200006**
3. Prodi : **Pendidikan Matematika**
4. Judul Skripsi : **Analisis Perbandingan Algoritma Cipher Permutasi dan RSA sebagai Pengamanan Pesan Teks: E-polynomial yang berhubungan dengan Kade Ternary**
5. Pembimbing : **Nur Hamid, Ph.D.**
6. Fokus Bimbingan : **Aspek Relevansi Judul, Landasan Teori, Aktualisasi Data, Ketajaman Analisis, Aspek bahasa, metodologi, kesesuaian antar bab, dan sistematika penulisan**

7. Konsultasi

TANGGAL	MATERI BIMBINGAN	KETERANGAN KONSULTASI/ARAHAN	PARAF
11/03/2023	konstruksi Grup	N. Hamid	
28/04/2023	menentukan E polynomial	N. Hamid	
04/05/2023	sistematika penulisan bab 4	N. Hamid	
06/05/2023	Orman: Ring dan Subring	N. Hamid	
12/05/2023	Polynomial Grmetis	N. Hamid	
20/05/2023	Pengelasan A.kh.	N. Hamid	

8. Bimbingan telah selesai pada tanggal 20 Mei 2023
Dosen Pembimbing I

Nur Hamid, Ph.D.

LAMPIRAN 2

PROGRAM PENELITIAN

Program Membangun Grup G Orde 2592

```
[1]: CF.<z> = CyclotomicField(12)
      w = z**4
      ii = z**3
      s3 = (2*w + 1)/ii
      SR(s3)
      print(SR(s3).expand())
      print(w)
      print((SR(z)**12).expand())

      sqrt(3)
      z^2 - 1
      1

[2]: T0 = diagonal_matrix([w, 1, 1])
      T1 = diagonal_matrix([1, w, 1])
      T2 = diagonal_matrix([1, 1, w])
      T3 = matrix(CF, [[1,0,0],[0,0,1],[0,1,0]])
      T4 = matrix(CF, [[0,1,0],[0,0,1],[1,0,0]])
      T5 = matrix(CF, [[1,1,1],[1,w,w**2],[1, w**2,w]])/s3

      T = [T0,T1,T2,T3,T4,T5]
      for i in T:
          print("berikut adalah matriks pembangun ke ", T.index(i))
          print(i)

[3]: G = MatrixGroup(T0,T1,T2,T3,T4,T5)
      G

[3]: Matrix group over Cyclotomic Field of order 12 and degree 4 with 6 generators

[4]: save(G, 'grup2592')

[5]: len(G) #orde grup

[5]: 2592

[6]: type(G)

[6]: <class 'sage.groups.matrix_gps.finitely_generated.FinitelyGeneratedMatrixGroup_g
      ap_with_category'>
```

Program Barisan Molien

```
[7]: mol = G.molien_series(return_series=False) #molien series
      save(mol, 'Molien2592')
      mol
```

```
[7]: (-t24 - 1)/(t60 - 2*t48 + t36 - t24 + 2*t12 - 1)
```

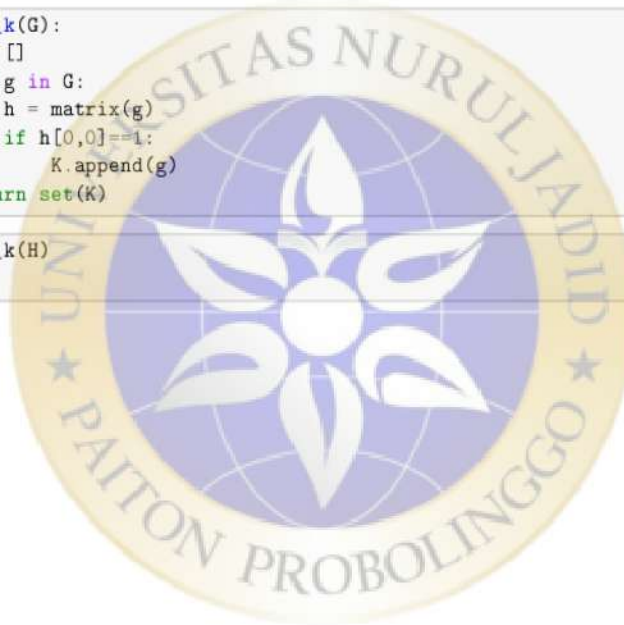
Program Grup Bagian K Orde 18

```
[1]: H = load('grup2592.sobj')
```

```
[2]: def sub_k(G):
      K = []
      for g in G:
          h = matrix(g)
          if h[0,0]==1:
              K.append(g)
      return set(K)
```

```
[3]: A = sub_k(H)
      len(A)
```

```
[3]: 18
```



Program Koset Kanan

```
[4]: def coset_element(K,g):
      return set([k*g for k in K])
```

```
[5]: def rep_cosets(G):
      Quot = []
      K = sub_k(G)
      Quot.append(K)
      Rep = [list(K)[0]]
      Gtemp = set(G)
      Gtemp = Gtemp.difference(K)
      k = 0

      for g in G:
          if g in Gtemp:
              Coset = coset_element(K,g)
              Gtemp = Gtemp.difference(Coset)
              Quot.append(Coset)
              Rep.append(g)

      k+=1
      print(k)
      return Rep
```

```
[6]: KOSET = rep_cosets(H)
```

Program E-polynomial

```
[7]: def epoly(Rep,X,k):
      ep = sum([(matrix(r)[0,:]*X)**k for r in Rep])
      return ep[0,0]
```

Program Mencari Index Setiap Degree n

```
[8]: def index_gen(n,r):
    L = [[n]]
    k = n-r
    m = r+0
    Lt=[]
    while k>=m:
        L1 = load('index/list'+str(k)+''.subj')
        L2 = load('index/list'+str(m)+''.subj')

        L3 = [l1+l2 for l1 in L1 for l2 in L2]
        print(k,m)
        m+=r
        k-=r
        Lt = Lt+L3
    for lt in Lt:
        lts = sorted(lt)
        if lts not in L:
            L.append(lts)
    save(L,'index/list'+str(n)+''.subj')
    return L
```

Program Dimensi

Program ini berbentuk fungsi, variabel dibuat sebanyak yang dibutuhkan. Penulis menggunakan fungsi ini untuk $n = 24$ dan menambah variabel sebagai fungsi baru untuk setiap degree.

```
[9]: def rank(m,n):
    M = m.coefficients()
    N = n.coefficients()
    M_MON = m.monomials()
    N_MON = n.monomials()

    M_len = len(M)

    M1 = []
    N1 = []
    if M_MON == N_MON:
        for i in range(M_len):
            X = m.monomial_coefficient(M_MON[i])
            M1.append(X)
            Y = n.monomial_coefficient(M_MON[i])
            N1.append(Y)

    mat = matrix([M1,N1]) #membuat matriks berdasarkan monomial yang sama
    rank = mat.rank()
    return rank
```


RIWAYAT HIDUP



NURUL JANNAH berdomisili di Desa Sambirampak Lor, Kecamatan Kotaanyar, Kabupaten Probolinggo. Penulis lahir di Probolinggo tanggal 05 Oktober 2001, anak pertama dari dua bersaudara, pasangan Bapak Sunandi dan Ibu Siti Fatimah. Pendidikan dasarnya ditempuh di SDN Sambirampak Lor selesai tahun 2013. Pendidikan menengah pertamanya di SMP Islam Paiton dengan berdomisili di Pondok Pesantren Islamiyah Syafi'iah, selesai tahun 2016. Pendidikan menengah atasnya ditempuh di MAN 1 Probolinggo dengan berdomisili di Pondok Pesantren Nurul Jadid, selesai tahun 2019. Penulis menempuh sarjana pendidikan matematika di Fakultas Sosial dan Humaniora Universitas Nurul Jadid Paiton Probolinggo.

Seiring dengan berbagai aktivitas yang penulis tekuni selama perkuliahan, penulis juga dapat menyelesaikan tugas akhir atau skripsi dengan judul "E-Polynomial yang Berhubungan dengan Kode Ternary" sebagai salah satu syarat untuk menyelesaikan studi pendidikan jenjang program Strata Satu (S1) Pendidikan Matematika.