Plagiarism Checker X Originality Report

Similarity Found: 6%

Date: Sunday, September 15, 2019
Statistics: 222 words Plagiarized / 3815 Total words
Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

Faktorisasi Polinomial Square-Free dan Bukan Square-Free atas Lapangan Hingga Juli Loisiana Butar-butar1, Ferdinand Sinuhaji2 1,2Matematika, Universitas Quality Berastagi, Indonesia julois.butrz@gmail.com <mailto:julois.butrz@gmail.com>1, sinuhajiferdinand@gmail.com <mailto:sinuhajiferdinand@gmail.com>2 INFO ARTIKEL ABSTRAK Riwayat Artikel: Diterima: 29-08-2019 Disetujui: 01-10-2019 Abstrak: Faktorisasi polinomial atas lapangan hingga memerlukan suatu metode yang tepat yakni algoritma untuk memproses faktorisasi polinomial.

Algoritma Faktorisasi Berlekamp merupakan salah satu metode terbaik dalam memfaktorisasi polinomial atas lapangan hingga . Polinomial atas lapangan terbagi dua kategori berdasarkan faktorisasinya, yaitu polinomial square-free dan polinomial bukan square-free. Polinomial square-free adalah polinomial yang tidak mempunyai faktor berulang. Sedangkan polinomial bukan square-free adalah sebaliknya.

Penelitian ini bertujuan untuk membuat suatu algoritma untuk menfaktorkan polinomial square-free dan bukan square-free atas lapangan hingga. Adapun yang menjadi referensi utama dalam penelitian ini adalah berdasarkan (Divas?n, Joosten, Thiemann, & Yamada, 2017). Namun, dibatasi hanya untuk polinomial square-free saja.

Metode yang digunakan adalah menggabungkan algoritma faktorisasi square-free dengan algoritma Berlekamp sehingga diperoleh algoritma baru. Pada bagian akhir penelitian adalah mengimplementasikan algoritma baru yang telah disusun. Abstract: Polynomial factorization over finite field needs a right method that is an algorithm to process polynomial factorization.

Algorithm Factorization Berlekamp is one of the best methods in factoring polynomials

over a finite field . Polynomials over field are divided into two category based on its factorization, namely square-free polynomials and not square-free polynomials. Square-free polynomials are polynomials polynomials that do not have repeated factors. When non square-free polynomials are the opposite.

This research aims to set an algorithm for factoring square-free polynomials and non-square-free polynomials over a finite field . The main reference in this research is based on (Divas?n, Joosten, Thiemann, & Yamada, 2017). However, it is restricted only to square-free polynomials. The method used is to combine the square-free factorization algorithm with the Berlekamp algorithm to obtain a new algorithm.

At the end of the research is to implement the new algorithm that has been compiled. Kata Kunci: Faktorisasi Polinomial; Lapangan Hingga; Algoritma Berlekamp; Polinomial Square-free Keywords: Polynomial Factorization; Finite Field, Berlekamp Algorithm; Square-free Polynomial. DOI: <https://doi.org/10.31764/jtam.v3i2.1079> ---------- u ---------- A.

LATAR BELAKANG Faktorisasi polinomial merupakan salah satu penunjang dasar dalam sistem komputer aljabar. Proses ini dibutuhkan dalam aljabar komputasi, lapangan hingga, serta aljabar linier. Salah satu metode yang digunakan untuk memfaktorkan polinomial tersebut adalah dengan metode Euclidean dan faktor persekutuan terbesar.

Namun, pada penelitian ini akan dibahas mengenai faktorisasi polinomial pada lapangan hingga ?p . Salah satu konsep yang sangat penting dalam pembahasan polinomial adalah akar-akar dari polinomial tersebut. Berdasarkan faktorisasi tak tereduksi, polinomial dibagi dalam dua jenis, yaitu polinomial yang faktorisasi tak tereduksinya tanpa perulangan atau disebut juga polinomial square-free, dan polinomial yang faktorisasi tak tereduksinya terdapat perulangan atau disebut juga polinomial bukan square-free.

Konsep polinomial square-free ini sama dengan polinomial separable (Saropah, 2012). Diberikan merupakan suatu polinomial atas berderajat dengan ??. Faktorisasi polinomial pada lapangan hingga sangat berguna dalam teori pengkodean, dan bahkan merupakan salah satu tahapan pendukung dalam menentukan grup Galois dari suatu polinomial (Butar-butar, 2018).

Berbeda dengan faktorisasi polinomial atas ring bilangan bulat, faktorisasi polinomial atas dapat difaktorisasi dengan cara mencoba-coba membagikan ? dengan polinomial tak tereduksi lain dengan derajat lebih kecil dari ?. Namun, faktorisasi dengan cara ini tidak efisien. Oleh karena itu, diperlukan suatu metode khusus berupa algoritma untuk

mencari faktor-faktor tak tereduksi dari polinomial tersebut.

Salah satu algoritma yang sangat tepat dan efisien dalam memfaktorkan polinomial adalah algoritma Berlekamp. Kebanyakan algoritma faktorisasi membatasi input hanya untuk polinomial square-free. Demikian juga halnya dengan Algoritma faktorisasi Berlekamp yang dibatasi untuk polinomial square-free (Divas´on, 2017).

Selain (Divas?n, Joosten, Thiemann, & Yamada, 2017), pada (Hanif, 2011) juga membahas algoritma Berlekamp dan beberapa contoh faktorisasi polinomial atas lapangan hingga serta implementasi algoritma tersebut. Pembatasan hanya untuk polinomial square-free inilah yang melatar-belakangi peneliti untuk membuat suatu algoritma faktorisasi yang tidak hanya untuk polinomial square-free tetapi juga untuk polinomial bukan square-free.

Aplikasi dari algoritma ini sangat diperlukan dalam kontruksi kode siklik berulang (Cao & Gao, 2014), (Batoul, Guenda, & Gulliver, 2015). Di lain sisi, (Lee, 2013) membahas tentang algoritma faktorisasi square-free yaitu suatu algoritma untuk mencari faktor square-free polinomial dan banyak dari perulangan faktor tersebut. Namun faktor-faktor tersebut belum tentu tak tereduksi.

Dengan menyusun langkah-langkah berdasarkan algoritma faktorisasi square-free dan algoritma Berlekamp diperoleh algoritma baru. Dengan demikian yang menjadi tujuan penelitian ini adalah menyusun algoritma faktorisasi polinomial untuk polinomial square-free dan polinomial bukan square-free dan kemudian mengimplementasikannya.
B. TINJAUAN PUSTAKA 1.

Algoritma Faktorisasi Berlekamp Dalam (Childs, 2009) dijelaskan hal yang melatarbelakangi dari Algoritma Berlekamp untuk memfaktorkan polinomial monik di berderajat didasarkan pada penentuan polinomial non-konstan berderajat lebih kecil atau sama dengan sehingga membagi habis sehingga kemudian diperoleh faktorisasi . Polinomial monik adalah polinomial yang pangkat tertinggi sama dengan satu (Mursyidah, 2017). Teorema 1. Diberikan polinomial monik square-free di berderajat .

Jika terdapat di polinomial berderajat berderajat dengan sedemikian hingga membagi habis , maka (1) adalah faktorisasi nontrivial dari di . Bukti. Berdasarkan yang diketahui membagi habis diperoleh dua fakta. Fakta pertama, berdasarkan Teorema Fermat, mempunyai akar sebanyak di , dengan . Kemudian berdasarkan Teorema Akar, terfaktorisasi di menjadi Dibentuk sehingga menghasilkan Sehingga merupakan perkalian dari polinomial-polinomial yang saling prima relatif di .

Fakta kedua, jika dan adalah polinomial saling prima relatif di , dengan lapangan, maka setiap polinomial di , Berdasarkan induksi, fakta ini digeneralisasi untuk kasus dari faktor persekutuan terbesar dari dan yang lebih dari dua pasangan faktor saling prima relatif. Karena membagi habis , diperoleh Karena dan saling prima relatif untuk dan , diperoleh Algoritma faktor persekutuan terbesar sangat menentukan dalam faktorisasi polinomial (Temnhurne & Sathe, 2016).

Pada faktorisasi dari Teorema 1, faktor persekutuan terbesar dapat ditentukan secara efisien dengan Algoritma Euclid (Iliev & Kyurkchiev, 2018) di . Untuk memfaktorkan di dengan menggunakan cara pada Teorema 1., maka dicari polinomial berderajat dengan , sedemikan hingga membagi habis . Ini dilakukan dengan membentuk dan menyelesaikan suatu sistem persamaan linear untuk memperoleh koefisien dari . Hal ini dijamin dalam teorema berikut ini.

Teorema 2. Diberikan polinomial monik square-free di berderajat . Diberikan di yang memenuhi Teorema 1 dan matriks berukuran dengan baris ke- merupakan vektor dari koefisien dari polinomial sisa untuk , maka koefisien-koefisien polinomial merupakan solusi persamaan linear homogen dengan adalah matriks identitas berukuran . Bukti.

Dimisalkan (2) dimana di merupakan koefisien-koefisien yang ditentukan. Berdasarkan Proposisi 12 ((Childs, 2009), hal 204), diperoleh Berdasarkan Teorema Fermat, untuk semua di sehingga (3) Untuk menentukan sisa dari dibagi , cari untuk (4) dengan .

Oleh karena itu, (5) sehingga persamaan (3) menjadi (6) Akibatnya, membagi habis jika dan jika membagi habis polinomial (7) Tetapi polinomial (7) berderajat , sehingga habis dibagi oleh jika dan hanya jika polinomial (7) merupakan merupakan poliomial nol di . Kondisi ini akan digunakan untuk menentukan koefisien-koefiesien dari .

Dengan kata lain, harus memenuhi (8) Jika koefisien-koefisien dari pada persamaan (8), diperoleh sebanyak persamaan-persamaan linier simultan di tidak diketahui dimana koefisien-koefisien di dari persamaan pada persamaan merupakan koefisien dari polinomial sisa . Solusi sistem persamaan ini menyajikan elemen-elemen yang merupakan koefisien-koefisien dari polinomial sedemikian hingga membagi habis .

Tranformasi bentuk persamaan (8) menjadi bentuk matriks. Dimisalkan matriks identitas , dimisalkan untuk setiap sehingga diperoleh matriks yang tak lain adalah matriks . Dengan demikian (9) merupakan matriks dengan setiap baris merupakan koefisien-koefisien dari polinomial sisa .

Akibatnya, dengan memeriksa komponen dari vektor diperoleh bahwa

koefisien-koefisien merupakan solusi persamaan homogen . Dengan menggunakan Teorema 1 dan Teorema 2 diperoleh teorema berikut. Teorema 3. Algoritma Faktorisasi Berlekamp Diberikan polinomial monik square-free di berderajat .

Diberikan matriks berukuran dengan baris ke- merupakan vektor dari koefisien dari polinomial sisa untuk . Diberikan adalah solusi dari dan . Jika berderajat , maka terdapat di sehingga dan mempunyai suatu faktor persekutuan berderajat lebih besar atau sama dengan 1. Bukti. Akan dibuktikan dengan menggunakan aturan kontradiksi. Asumsikan untuk semua di dan tidak mempunyai faktor persekutuan berderajat lebih kecil dari 1.

Ini berarti, dengan . Akibatnya, Karena polinomial berderajat yang tidak sama dengan nol, maka Akibatnya hal ini kontradiksi Teorema 1 Sehingga berdasarkan Teorema 1 diperoleh untuk semua di merupakan polinomial berderajat dan sedemikian hingga tidak membagi habis . Karena di , maka tidak membagi habis . Ini berarti, .

Ini berarti, tidak membagi habis . Akibatnya, tidak membagi habis polinomial (10) Kemudian berdasarkan Teorema Akar, diperoleh Karena dan saling prima untuk dan , maka diperoleh Akibatnya, membagi habis jika dan jika membagi habis polinomial (10).

Karena polinomial (10) berderajat dan tidak habis dibagi oleh , maka polinomial (10) merupakan bukan merupakan poliomial nol di . Hal ini kontradiksi dengan diketahui . Dengan demikian, asumsi salah. Ini berarti, terdapat di sehingga dan mempunyai suatu faktor persekutuan berderajat lebih besar atau sama dengan 1.

Dari proses pembuktian Teorema 3 dapat disimpulkan bahwa bagian terpenting adalah menyelesaikan sistem persamaan linear homogen dengan adalah vektor dengan panjang , adalah matriks berukuran , dan adalah vektor nol dengan panjang . Sistem persamaan linear homogen tersebut dapat juga diselesaikan dengan cara menyelesaikan bentuk sistem persamaan linear homogen (11) Oleh karenanya, sistem persamaan linear homogen pada persamaan (11) berhubungan dengan mencari ruang null dari matriks .

Karena matriks atas lapangan hingga, maka peneliti akan menggunakan hasil pada (Abdel-Ghaffar, 2012) sebagai referensi untuk menentukan solusi sistem persamaan linear homogen. Referensi ini dipakai dalam proses komputasi. Suatu polinomial tak tereduksi atas juga dapat dideteksi dengan algoritma Berlekamp. Ini dapat menjadi suatu bagian untuk mengkontruksi suatu polinomia tak tereduksi atas lapangan hingga (Couveignes & Lercier, 2013).

Namun, algoritma Berlekamp tidak dapat dipakai untuk lapangan perluasan untuk

sehingga diperlukan algoritma lain yang lebih tepat (Divasón, Joosten, Thiemann, & Yamada, 2019). C. METODE PENELITIAN Bagian utama yang menjadi bagian dasar penelitian ini adalah Algoritma faktorisai Berlekamp yang berlaku untuk polinomial square-free atas .

Algoritma Pembagian pada polinomial, Metode Euclidean dan Faktor Persekutuan Terbesar untuk memfaktorisasi polinomial (Oktafia, Gemawati , & Endang, 2014) adalah metode yang dipakai untuk menyelesaikan salah satu langkah dari Algoritma faktorisai Berlekamp. Metode Operasi baris elementer yang telah diformalisasikan dalam (Aransay & Divasón, 2016) berlaku untuk lapangan hingga.

Solusi Sistem Persamaan Linear Homogen atas lapangan hingga dengan menggunakan Metode Operasi baris elementer atas lapangan hingga (Abdel-Ghaffar, 2012) sebagai sistem persamaan linear homogen. Dengan metode operasi baris elementer dapat juga ditemukan banyak rank dari suatu matriks atas lapangan hingga (Grout, 2010) yang akibatnya banyak ruang null pun dapat diperoleh. Pembatasan Algoritma Berlekamp hanya untuk polinomial square-free mengharuskan penelitian ini menggunakan metode tambahan lain.

Metode ini adalah algoritma untuk menentukan polinomial faktorisasi square-free dan banyak polinomial tersebut dari suatu polinomial yang disebut dengan Algoritma square-free factorization (Lee, 2013). Selanjutnya, peneliti membuat suatu algoritma baru berdasarkan Algoritma square-free factorization dan Algoritma Berlekamp. Kemudian sebagai bagian akhir penelitian, peneliti telah mengimplementasi algoritma tersebut dalam program Matlab 2012a dimana setiap langkah yang memerlukan penyelesaian secara khusus dapat dijalankan dengan fungsi yang sudah ada di Matlab 2012a atau peneliti membuat fungsi khusus untuk langkah itu.

Salah satu fungsi Matlab yang harus diimplementasikan adalah operasi baris elementer atas lapangan . Dalam operasi baris elementer merupakan metode eliminasi Gauss-Jordan merupakan salah satu langkah dan proses ini telah diimplementasikan dalam Matlab (Irwan, 2017). Lebih jelas mengenai metode penelitian, perhatikan Gambar 1 berikut. Gambar 1. Diagram Alir Penelitian. D. HASIL DAN PEMBAHASAN 1.

Algoritma Faktorisasi Square-Free Kebanyakan algoritma faktorisasi polinomial atas lapangan hingga mengharuskan polinomial input tidak memiliki faktor berulang atau disebut juga polinomial square-free (Lee, 2013). Salah satunya adalah Algoritma Berlekamp yang dibatasi hanya untuk polinomial square-free saja. Berikut ini merupakan algoritma Berlekamp yang terdapat di (Divas?n, Joosten, Thiemann, & Yamada, 2017). Algoritma 1.

Faktorisasi Berlekamp Input: polinomial square-free atas lapangan berderajat . Output: Konstanta dan himpunan polinomial monik tak tereduksi sehingga 1. Bentuk 2. adalah koefisien pangkat tertinggi dari . Bentuk 3. Hitung matriks Berlekamp untuk , dengan baris ke- adalah vektor dari koefisien polinomial untuk . 4. Hitung dimensi dan basis dari ruang null , dimana adalah matriks identitas berukuran . 5. Untuk setiap vektor bentuk menjadi polinomial . 6. Bentuk , , . 7.

Jika , kembali dan . 8. Pilih dan perbarui . Perbarui 9. Perbarui dan 10. Jika ditemukan polinomial tak tereduksi sebanyak di , maka pindahkan ke dan perbarui . Kembali ke langkah 7. Jika tidak ke langkah 10 11. Hasil adalah konstanta dan himpunan . 12. Selesai. Namun, ada suatu algoritma untuk menentukan faktorisasi square-free dari polinomial input dan banyak pergandaan dari faktorisasi tersebut. .

Algoritma ini disebut dengan Algoritma square-free factorization. Berikut ini adalah Algoritma square-free factorization dengan mengubah bentuk algoritma dari yang dibahas (Lee, 2013). Algoritma 2. Algoritma square-free factorization Input: polinomial monik atas lapangan berderajat . Output: . 1. Bentuk . 2. Tentukan atas 3. Jika ,maka polinomial menjadi .

Kemudian . Lanjut ke langkah 23. Jika tidak, lanjut ke langkah 4. 4. Hitung atas . 5. Tentukan polinomial atas dan . 6. Jika , maka atas . Lanjut ke langkah 7. Jika tidak, lanjut ke langkah 8. 7. Jika , maka . Lanjut ke langkah 22. Jika tidak lanjut ke langkah 8. 8. Tentukan atas . 9. . 10. Hitung atas . 11. Tentukan atas . 12. Jika , maka . . Lanjut ke langkah 14. Jika tidak lanjut ke langkah 13. 13. Bentuk . 14. Ubah . 15. Ubah atas . 16.

Jika , bentuk polinomial yang dibentuk dari hasil kali semua polinomial di dan perulangannya. Lanjut ke langkah 17. Jika tidak, lanjut ke langkah 19. 17. Ubah atas , merupakan polinomial bukan square-free yang hanya memiliki satu faktorisasi. 18. Tentukan atas . Hitung atas . Hitung atas 19. Bentuk . Lanjut ke langkah 22. 20. . Jika kembali ke langkah 10. Jika tidak, lanjut ke langkah 21. 21. Tentukan atas 22. Jika ,maka polinomial menjadi . Kemudian .

Lanjut ke langkah 23. 23. Selesai. Dengan menggabungkan Algoritma 1 dan 2, selanjutnya dibentuk suatu algoritma baru yang menjadi penyelesaian dari faktorisasi untuk polinomial square-free dan bukan square-free atas yang akan dibahas pada bagian berikutnya. 2.

Algoritma Faktorisasi Polinomial Square-Free dan bukan Square-Free Pada bagian ini, dibentuk algoritma baru yang merupakan gabungan Algoritma 1 dan 2 sebagai

penyelesaian dari faktorisasi untuk polinomial square-free dan bukan square-free atas . Algoritma 3. Algoritma Faktorisasi Polinomial Square-Free dan bukan Square-Free Input: polinomial atas lapangan berderajat . Output: adalah koefisien pangkat tertinggi . . 1.

Bentuk . 2. adalah koefisien pangkat tertinggi dari . 3. Bentuk ulang polinomial menjadi . 4. Tentukan yakni himpunan semua faktorisasi square-free dari dengan menggunakan Algoritma 2. 5. Tentukan adalah banyaknya faktorisasi square-free dari dari langkah 4. 6. 7.

Tentukan yang merupakan himpunan faktorisasi tak tereduksi polinomial square-free dari himpunan anggota ke- dengan menggunakan Algoritma 1. 8. Jika maka Lanjut ke langkah 10. Jika tidak, lanjut ke langkah 9. 9. Tentukan yang merupakan banyaknya himpunan dari dengan . 10. Untuk 11. . Jika ,kembali ke langkah 7. 12. Selesai. Konsep dasar dari algoritma 3 adalah mencari faktor square-free dari polinomial dan banyak perulangannya.

Selanjutnya, setiap faktor square-free tersebut difaktorisasi lagi dengan menggunakan algoritma Berlekamp. Untuk lebih jelas mengenal Algoritma 3, perhatikan contoh berikut ini. Contoh 1. Faktorisasi polinomial berikut atas . Solusi: Karena koefisien pangkat tertinggi adalah 2, maka polinomial menjadi Turunan pertama terhadap Kemudian, Lalu diperoleh dan derajat dari , .

Karena , maka kemudian untuk diperoleh: Iterasi Lalu diperoleh atas . Karena , maka bukan faktorisasi yang dimaksud. Kemudian ubah dan atas sehingga Ubah . Iterasi Lalu diperoleh Karena , maka Kemudian ubah dan atas sehingga Karena , maka bentuk Ubah atas sehingga Kemudian . Lalu sehingga diperoleh atas . Akibatnya, faktorisasi square-free dari Banyaknya faktorisasi square-free dari adalah .

Langkah selanjutnya adalah menentukan faktorisasi tak tereduksi dari polinomial di . Karena tak tereduksi, cukup mencari faktorisasi dari dengan Algoritma 1. Matriks Berlekamp adalah sehingga Dengan operasi baris elementer atas diperoleh Dan ruang null dari adalah sehingga dan . Selanjutnya, proses cukup untuk saja. Kemudian dihitung untuk . , , , , dan . Sehingga . Karena semua polinomial di tak tereduksi, maka proses untuk Algoritma 1 selesai.

Karena , maka Karena tak tereduksi, maka himpunan faktorisasi tak tereduksi adalah dengan koefisien pangkat tertinggi adalah 2. 3. Implementasi Algoritma sebagai Fungsi Matlab. Penelitian ini menggunakan Matlab 2012a untuk mengimplementasikan algoritma-algoritma yang telah dibahas sebelumnya. Alasan dari penelitian ini memilih Matlab adalah karena merupakan program yang cukup umum dipakai dan dapat

sebagai produk berupa bahan ajar pada mata kuliah Pemrograman (Syaharuddin & Mandailina, 2017). Karena implementasi ini sangat berguna dalam mata kuliah Teori Pengkodean.

Salah satunya dalam mengkontruksi kode siklik (Ding, 2017). Pada dasarnya, ada beberapa algoritma yang bukan merupakan fungsi bawaan Matlab 2012a namun tidak dapat peneliti tampilkan. Fungsi-fungsi Matlab itu adalah sebagai berikut. · gfinv(a,p) : untuk mencari invers perkalian dari bilangan tak nol .

· gfgauss(A,p) : untuk menentukan bentuk eliminasi gauss dari matriks atas . · nullgf(A,p) : untuk menentukan basis ruang null dari matriks atas . · vektorpol(v) : untuk mengubah suatu bentuk vector dari koefisien polinomial menjadi polinomial · gfgcd(f,g,p) : untuk mencari faktor persekutuan terbesar dari dua polinomial dan atas . · doubfac(f,p) : untuk Mendeteksi Polinomial yang faktorisasinya hanya satu polinomial square-free dan pergandaannya.

Berikut ini beberapa fungsi matlab yang lain yang telah implementasikan. 1. Fungsi Matlab untuk Mendeteksi Faktorisasi Square-free dari Polinomial atas function [Fs]=sf(f,p) %f polinomial monik S=[]; R=[]; d=length(f)-1; g=mod(polyder(f),p)%b syms x pf=vektorpol(fliplr(f)); if g==0 pfp=subs(pf,x,x^(1/p)); kfp=sym2poly(pfp) ; Fs=[S; pfp,((length(f)-1)/(length(kfp)-1))]; else k=gfgcd(f,g,p)%k k=mod(gfinv(k(length(k)),p)*k,p); [h,s]=gfdeconv(fliplr(f),k,p);%k lh=length(h)-1; if rem(d,lh)==0 ph=mod(vektorpol(h)^(d/lh),p); kph=sym2poly(ph);%b if ph==pf [Fs]=[S;vektorpol(h),(d/lh)]; end else w=fliplr(h); w1=gfinv(w(1),p); w=mod(w1*w,p); %menidentifikasi semua faktor w dw=mod(polyder(w),p)%b gw=gfgcd(w,dw,p); i=1; while length(w)~=1 y=gfgcd(fliplr(k),w,p); [fac,s2]=gfdeconv(fliplr(w),y,p); f1=gfinv(fac(length(fac)),p); fac=mod(f1*vektorpol(fac),p); if fac==1 fac=[]; S=[S;fac]; R=[R (fac)^i]; else S=[S;fac,i]; R=[R (fac)^i]; end w=fliplr(y); [k,s3]=gfdeconv(k,y,p); k=mod(gfinv(k(length(k)),p)*k,p); if k==1 rf=mod(expand(prod(R)),p); rf=sym2poly(rf); k=gfdeconv(fliplr(f),fliplr(rf),p); [T]=doubfac(fliplr(k),p); S=[S;T]; break else i=i+1; end end syms x k=fliplr(k); dk=mod(polyder(k),p); pk=vektorpol(fliplr(k)); if dk==0 fk=subs(pk,x,x^(1/p)); vfk=sym2poly(fk); Fk=[fk^((length(k)-1)/(length(vfk)-1))]; Fs=[S; fk,((length(k)-1)/(length(vfk)-1))]; else Fs=S; end end end 2.

Fungsi Matlab untuk menentukan Matriks Berlekamp dari Polinomial atas % f koefisien polinomial dan p bilangan prima function Bf=matriksberl(f,p) f=gfinv(f(1),p)*f; f=mod(f,p); d=length(f)-1; t=p*d+1; Bf=[]; i=1; while i<=d h=zeros(1,t); h(p*((d-i)+1)+1)=1; [r,s]=deconv(h,f); Bf=[Bf;s]; i=i+1; end Bf=mod(Bf,p); Bf=Bf(:,t-d+1:t); Bf=fliplr(Bf); 3.

Fungsi Matlab untuk mendeteksi Faktor Persekutuan Terbesar dari Polinomial dan Polinomial yang dibentuk dari Ruang Null Matriks Berlekamp function [b,Fc]=fbs(f,p) %masukkan koef polinomial dari pangkat tertinggi b=f(1); c=gfinv(b,p); %koefisien dari f fo=mod(c*f,p); % bentuk monik % menghitung matriks berlekamp Bf=matriksberl(fo,p); Bfi=mod(Bf-eye(size(Bf)),p); Bfi=Bfi'; % ruang null dari (Bf-I)^T Z = nullgf(Bfi,p); Z=Z'; [u,v]=size(Z); H=fliplr(Z(2:u,:)); Fc=[]; Gc=[]; for i=1:u-1 j=0; while j<p hj=H(i,:)-[zeros(1,v-1) j]; hj=mod(hj,p); gc=gfgcd(fo,hj,p); if length(gc)==1 G=[]; Gc=[Gc,G]; else G=vektorpol(gc); if gc(length(gc))==0 G=G; else G=mod(gfinv(gc(length(gc)),p)*G,p); end Gc=[Gc,G]; end Fc=union(Fc,Gc); j=j+1; end end if length(Fc)>1 for k=1:length(Fc) kfc1=sym2poly(Fc(k)); r=k+1; while r<=length(Fc) kfc2=sym2poly(Fc(r)); gkf=gfgcd(kfc1,kfc2,p); if length(gkf)==1 Hk=[]; Fc=[Hk Fc]; else if length(kfc1)>length(kfc2) hg=gfdeconv(fliplr(kfc1),gkf,p); Hk=vektorpol(hg); Fc=[Hk Fc]; else hg=gfdeconv(fliplr(kfc2),gkf,p); Hk=vektorpol(hg); Fc=union(Hk,Fc); end end r=r+1; end end elseif length(Fc)==1 th=gfdeconv(fliplr(f),fliplr(sym2poly(Fc)),p); th=vektorpol(th); Fc=union(th,Fc); else Fc=[]; end 4.

Fungsi Matlab untuk mendeteksi faktorisasi tak tereduksi Algoritma Berlekamp function [kf,If]=berlekamploop(f,p) %masukkan polinomial dari pangkat tertinggi kf=f(1); [kf,Fc]=fbs(f,p); if length(Fc)==0 f=fliplr(f); If=vektorpol(f); else d=length(f)-1; If=[]; for i=1:d^p f1=sym2poly(Fc(1)); [b1,fc]=fbs(f1,p); fc=fc; if length(fc)==0 If=union(If,Fc(1)); Fc=setdiff(Fc,Fc(1)); else Fc=setdiff(Fc,Fc(1)); Fc=union(fc,Fc); end if length(Fc)==0 break else continue end end end 5.

Fungsi Matlab untuk Algoritma 3 function [kf,Fac]=gabungan(f,p) Fac=[]; kf=f(1); d=length(f)-1; c=gfinv(kf,p); %koefisien dari f f=mod(c*f,p); Fs=sf(f,p); [m,n]=size(Fs); i=1; while i<=m fs=sym2poly(Fs(i,1)); V=ddfp(fs,p); vf=sym2poly(V(1)); dvf=doufac1(vf,p); fv=sym2poly(dvf(1)); vf=sffmy(fv,p); vf=sym2poly(vf(1)); [b1,Vf]=berlekamploop(vf,p); if length(Vf)==1 Fac=[Fac;Vf(1),V(1,2)*Fs(i,2)*dvf(1,2)]; else k=length(Vf); for j=1:k Fac=[Fac;Vf(j),V(1,2)*Fs(i,2)*dvf(1,2)]; end end i=i+1; end Adapun untuk fungsi yang dijalankan untuk menyelesaikan persoalan adalah fungsi Matlab gabungan(f,p).

Untuk menjalankan fungsi-fungsi tersebut dengan Matlab setiap fungsi disimpan dengan nama {nama_fungsi}.m dalam satu folder. Berikut hasil running program yang ditampilkan pada jendela command window dari Contoh 1. f=[2 1 0 1 3 2 0 2 3 2 2 0 1 4]; p=5; [kf,Fac]=gabungan(f,p) Fs = [x^5 + 2*x^4 + 4*x^3 + 2*x^2 + 4*x + 1, 2] [x + 3, 3] Bf = 1 0 0 0 0 4 1 3 1 3 3 4 3 2 4 1 2 0 0 1 0 2 3 1 0 Z = 1 0 0 2 0 2 0 1 0 0 kf = 2 Fac = [x^2 + 2, 2] [x^3 + 2*x^2 + 2*x + 3, 2] [x + 3, 3] Berdasarkan hasil keluaran running program diperoleh hasil akhir himpunan pasangan terurut Fac dengan tiga anggota.

Untuk setiap pasangan terurut tersebut adalah urutan pertama merupakan faktorisasi

polinomial tak tereduksi dan urutan kedua adalah penggandaan polinomial tersebut untuk setiap pasangan terurut. Hasil ini sama dengan solusi contoh 1. E. SIMPULAN DAN SARAN Algoritma Berlekamp merupakan salah satu metode untuk menfaktorisasi polinomial atas lapangan , namun hanya dibatasi untuk polinomial square-free.

Dengan menggabungkan Algoritma Berlekamp dan Algoritma Faktorisasi square-free diperoleh suatu algoritma baru untuk faktorisasi polinomial square-free dan bukan square-free atas . Untuk penelitian lanjutan, peneliti menyarankan untuk faktorisasi polinomial square-free dan bukan square-free atas dengan . UCAPAN TERIMA KASIH Penelitian ini didanai oleh Direktorat Riset dan Pengabdian Masyarakat (DPRM) Kementerian Riset Teknologi dan Pendidikan Tinggi melalui Program Penelitian Dosen Pemula tahun pendanaan 2019.

Terima kasih juga untuk LPPM Universitas Quality Berastagi yang telah memfasilitasi peneliti hingga penelitian ini selesai. REFERENSI Abdel-Ghaffar, K. (2012). Counting Matrices over Finite Field Having a Given Number of Rows of Unit Weight. Linear Algebra and its Applications, 436, 2665-2669. Aransay, J., & Divasón, J. (2016). Formalisation of the computation of the echelon form of a matrix in Isabelle/HOL. Formal Aspects of Computing, 28, 1005-1026. Batoul, A., Guenda, K.,

& Gulliver, T. A. (2015). Repeated-Root Isodual Cyclic Codes over Finite Fields. Springer International Publishing Switzerland 2015, 119-132. Butar-butar, J. L. (2018). Menentukan Grup Galois Pada Polinomial Rasional Dengan Pengembangan Metode Stauduhar. Jurnal Curere, 2(2), 184-193. Cao, Y., & Gao, Y. (2014). Repeated root cyclic F_q-linear codes over F_(q^l). Journal Finite Fields and Their Applications, 31, 202-227. Childs, L. N. (2009).

A Concrete Introduction to Higher Algebra Third edition. New York: Springer. Couveignes, J.-M., & Lercier, R. (2013). Fast Construction of Irreducible Polynomials over Finite Fields. Israel Journal of Mathematics, 194, 77-105. Ding, C. (2017). A sequence construction of cyclic codes over finite fields. Cryptography and Communications, 10(2), 319-341. Divasón, J., Joosten, S., Thiemann, R., & Yamada, A.

(2019). A Verified Implementation of the Berlekamp-Zassenhaus Factorization Algorithm. Journal of Automated Reasoning, 63, 1-37. doi:https://doi.org/10.1007/s10817-019-09526-y Divas?n, J., Joosten, S., Thiemann, R., & Yamada, A. (2017). A Formalization of the Berlekamp-Zassenhaus Factorization Algorithm. CPP 2017 Proceeding of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, (pp. 17-29). Grout, J. (2010). The minimum rank problem over finite fields.

Electronic Journal of Linear Algebra, 20, 691-716. Hanif, S., & Imran, M. (2011). Factorization Algorithms for Polynomials over Finite Fields. Linnæus University, Departement of Mathematics. Retrieved April 19, 2019, from http://www.diva-portal.org/smash/get/diva2:414578/FULLTEXT01.pdf Hanif, Sajid. (2011). Factorization Algorithms for Polynomials over Finite Fields. Master Thesis, Departement of Mathematics. Iliev, A., & Kyurkchiev, N. (2018).

A Note on Euclidean and Extended Euclidean Algorithms for Greatest Common Divisor for Polynomials. International Journal of Pure and Applied Mathematics, 118(3), 713-721. Irwan, M. (2017). Pengantar Matlab Untuk Sistem Persamaan Linear. Jurnal MSA, 5(2), 48-53. Lee, M. M.-D. (2013). Factorization of multivariate polynomials. Dissertation, Technischen Universitat Kaiserslautern, Mathematics, Kaiserslautern.

Retrieved July 28, 2019, from https://d-nb.info/1036637972/34 Mursyidah, H. (2017). Algoritma Polinomial Minimum untuk Membentuk Matriks Diagonal dari Matriks Persegi. Aksioma Jurnal Pendidikan Matematika FKIP Univ. Muhammadiyah Metro, 6(2), 282-293. Oktafia, R., Gemawati , & Endang. (2014). Faktorisasi Polinomial Aljabar dengan Menggunakan Metode Euclidean dan Faktor Persekutuan Terbesar. Jurnal Online Mahasiswa FMIPA UNRI, 1-5. Saropah. (2012).

Akar-Akar Polinomial Separable Sebagai Pembentuk Perluasan Normal Pada Ring Modulo. Jurnal Cauchy, 2(3), 148-153. Syaharuddin, & Mandailina, V. (2017). Pengembangan Modul Pemrograman Komputer Berbasis Matlab. Jurnal Teori dan Aplikasi Matematika, 1(1), 1-4. Temnhurne, J., & Sathe, S. R. (2016). New Modified Euclidean and Binary Greatest Common Divisor Algorithm., 62(6), 852-858. IETE Journal of Research, 62(6), 852-858.

INTERNET SOURCES:

---

http://seminar.uny.ac.id/semnasmatematika/sites/seminar.uny.ac.id.semnasmatematika/files/banner/A%20-%207.pdf
<1% -
https://skripsi-skripsiun.blogspot.com/2014/10/skripsi-industrial-engineering-analisis_75.html
<1% -
https://www.researchgate.net/publication/325448319_ALGORITMA_POLINOMIAL_MINIMUM_UNTUK_MEMBENTUK_MATRIKS_DIAGONAL_DARI_MATRIKS_PERSEGI
<1% - https://id.scribd.com/doc/105544007/Faktor-Persekutuan-Terbesar
<1% - https://jurnalpengairan.ub.ac.id/index.php/jtp/article/viewFile/119/119
<1% - http://a-research.upi.edu/operator/upload/s_mat_055914_chapter3.pdf
<1% -
https://caraexcelpowerpointmsword.blogspot.com/2016/07/rumus-excel-if-cara-menggunakan-fungsi-if-di-microsoft-excel.html
<1% -
https://ngadiyonopendmtk.blogspot.com/2015/02/representasi-graph-dan-beberapa-graph.html
1% -
https://www.pinterpandai.com/aljabar-linear-matriks-homogen-diagonal-transpos-ruang-euklidian-soal-jawaban/
<1% - http://eprints.unm.ac.id/5182/1/ISI.docx
<1% -
https://alhanisberbagiilmu.blogspot.com/2015/06/contoh-skripsi-jurusan-tehnik-elektro.html
<1% - https://docplayer.info/32488543-Rank-matriks-adjacency-dari-graf-skripsi.html
<1% - https://sayfudinblogz.blogspot.com/2014/01/
<1% - https://www.proprofs.com/quiz-school/story.php?title=mji2nju1oqx1e6
<1% -
https://www.sederet.com/tutorial/vocabulary-part-of-body-bagian-bagian-tubuh/
<1% - https://www.seputarforex.com/belajar/forex/
<1% - http://www.klippansbk.se/pdf/ladyopen2013_statistik.pdf
<1% -
https://www.coursehero.com/file/p2dgoe1/0-0-0-0-0-0-0-0-0-0-x-2-0-1-0-3-0-4-0-5-0-1-x-3-0-9-8-7-6-5-4-3-2-1-x-4-0-0-1-2/
<1% -
http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Makalah2015/Makalah_IF221_Strategi_Algoritma_2015_004.pdf
<1% -
https://ristekdikti.go.id/wp-content/uploads/2015/11/Buku-Panduan-Pelaksanaan-Penelitian-dan-Pengabdian-kepada-Masyarakat-Edisi-XII.pdf

<1% - https://www.deepdyve.com/lp/elsevier/counting-matrices-over-finite-fields-having-a-given-number-of-rows-of-lxyZotvNfw
<1% - https://link.springer.com/chapter/10.1007%2F978-3-662-04053-9_1
<1% - http://www.ams.org/journals/mcom/2019-88-317/S0025-5718-2018-03363-4/
<1% - https://arxiv.org/pdf/1802.01336.pdf
<1% - https://www.researchgate.net/publication/251299070_Spectral_analysis_of_a_class_of_Schrodinger_difference_operators
<1% - https://link.springer.com/chapter/10.1007/978-3-030-01560-2_10
<1% - https://www.researchgate.net/publication/324532523_A_Note_on_Euclidean_and_Extended_Euclidean_Algorithms_for_Greatest_Common_Divisor_for_Polynomials
<1% - http://ejournal.uin-malang.ac.id/index.php/Math/article/download/3124/4989
<1% - https://www.researchgate.net/post/Any_hint_on_how_to_approximate_the_inverse_of_a_diagonal_dominant_matrix
<1% - https://acadsol.eu/dsa/28/1/8