Penerapan Konsep Ruang Vektor Umum dalam Efisiensi Algoritma Enkripsi *Elliptic Curve Cryptography* (ECC)

Renuno Yuqa Frinardi 13524080[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13524080@std.stei.itb.ac.id*, [2]*renunofrinardi@gmail.com*

***Abstract*— Makalah ini memberikan penjelasan mengenai perbandingan antara dua pendekatan kalkulasi dalam kriptografi *Elliptic Curve Cryptography* (ECC). Ditunjukkan bagaimana pendekatan koordinat proyeksi berbobot memiliki waktu pemrosesan yang lebih efisien dibanding dengan pendekatan koordinat biasa, terutama pada kalkulasi dengan nilai yang besar. Ini menunjukkan mengapa banyak implementasi ECC di dunia nyata menggunakan pendekatan koordinat proyeksi berbobot.**

***Keywords*—Elliptic Curve Cryptography (ECC), Cryptography, Ruang Vektor Umum, Proyeksi Berbobot, Group Theory, Galois Field, Aljabar Linear.**

I. Pendahuluan

Teknologi sudah berkembang dengan pesat di era ini. Beragam aktivitas manusia bisa dilakukan secara online dan dari mana saja. Memberikan kemudahan dalam melaksanakan kegiatan. Aktivitas online ini tentunya paling sering melalui internet, tempat di mana banyak informasi diterima ataupun dikirim antar orang dari berbagai belahan dunia.

Informasi yang dikirim tentunya memiliki beragam jenis. Ada informasi personal/privat ataupun informasi publik yang bisa diakses oleh siapapun. Informasi personal/privat tentunya menjadi informasi yang harus ketat dijaga penyebarannya, di mana hanya orang tertentu saja yang bisa mendapatkan informasi tersebut.

Walaupun begitu, banyak pihak tidak berwenang yang berusaha mendapatkan informasi privat tersebut demi kepentingannya sendiri. Informasi ini bisa didapat oleh orang tidak bertanggung jawab dengan beragam cara. Salah satunya adalah saat terjadi pertukaran atau transfer informasi melalui internet.

Menanggapi hal ini, kriptografi menjadi hal umum yang diterapkan dalam penyebaran data di internet. Dengan demikian, data yang terambil oleh orang tidak bertanggung jawab bukan merupakan data asli yang bisa langsung dibaca, tetapi sebuah informasi yang sudah terenkripsi.

Di dunia ini, banyak jenis kriptografi yang digunakan dalam enkripsi. Salah satunya adalah *Elliptic Curve Cryptography* (ECC). Sebuah kriptografi asimetris yang sering digunakan dalam dunia modern, dari mulai dalam cara kerja *Cryptocurrency* sampai dengan enkripsi pada protokol network *Secure Shell* (SSH).

Tentunya ECC ini memiliki cara kalkulasinya tersendiri. Perhitungan pada ECC ini bersifat banyak dan repetitif. Melihat hal ini, banyak pendekatan yang bisa dilakukan ke dalam implementasi enkripsi ini. Salah dua dari pendekatan ini adalah pendekatan affine dan *projective coordinates* (transformasi ruang vektor). Makalah ini bertujuan untuk melakukan perbandingan kedua pendekatan perhitungan dalam hal efisiensi waktu. Diharapkan perbandingan ini dapat menilai dan menunjukkan pendekatan apa yang lebih efisien dari segi waktunya.

II. Dasar Teori

A. Vektor di Ruang Euclidean

Sebuah ruang Euclidean berdimensi-n ditulis dengan bentuk $\mathbb{R}^n$, adalah himpunan semua n-tupel terurut dari bilangan riil. Apabila n adalah sebuah bilangan bulat positif, maka sebuah vektor **v** di dalam ruang $\mathbb{R}^n$ dapat dinyatakan sebagai urutan dari bilangan riil $(v_1, v_2, ..., v_n)$.

Vektor-vektor di ruang euclidean didefinisikan melalui dua operasi dasar, yaitu penjumlahan vektor dan perkalian dengan skalar. Misalkan $u = (u_1, u_2, ..., u_n)$ dan $v = (v_1, v_2, ..., v_n)$ adalah vektor-vektor di $\mathbb{R}^n$, dan didefinisikan k sebuah skalar riil, sehingga:

1) **Penjumlahan**: $u + v = (u_1 + v_1, u_2 + v_2, ..., u_n + v_n)$
2) **Perkalian Skalar** $ku = (ku_1, ku_2, ..., ku_n)$

Vektor-vektor dalam ruang euclidean juga memiliki definisi jarak dan panjang (norma vektor) yang baku. Panjang atau norma dari suatu vektor v di $\mathbb{R}^n$, dinotasikan dengan $\|v\|$, dan didefinisikan menggunakan *Euclidean norm* sebagai berikut:

$$\|v\| = \sqrt{v_1^2 + v_2^2 + ... + v_n^2}$$

Definisi norma di atas membantu mendifinisikan bahwa jarak antara dua vektor u dan v adalah $\|u - v\|$

B. Vektor di Ruang Umum

Ruang vektor secara umum dapat didefinisikan sebagai himpunan objek-objek yang dilengkapi dengan dua operasi di dalam himpunan tersebut, yakni:

1) Operasi penjumlahan objek-objek
2) Operasi perkalian sebuah objek dengan skalar

Misal V adalah sebuah himpunan objek-objek dengan operasi penjumlahan antara objek dan perkalian dengan skalar,

agar V terdefinisi sebagai ruang vektor, operasi penjumlahan antara objek dan perkalian dengan skalar harus memenuhi 6 aksioma berikut:

1) **Tertutup** (closure): Operasi penjumlahan dan perkalian skalar harus menghasilkan vektor yang merupakan anggota dari V. Jadi, untuk semua $u, v \in V$ dan skalar k, maka harus memenuhi
$$u + v \in V$$
$$ku \in V$$
2) **Komutatif**: Untuk semua $u, v \in V$, maka harus memenuhi $u + v = v + u$
3) **Asosiatif**: Untuk semua $u, v, w \in V$, maka harus memenuhi $u + (v + w) = (u + v) + w$
4) **Identitas**: Untuk semua $u \in V$, terdapat sebuah elemen identitas (vektor) 0 dan skalar 1 sedemikian sehingga harus memenuhi
$$u + 0 = 0 + u = u$$
$$1u = u$$
5) **Balikan (invers) atau negatif**: Untuk setiap $u \in V$, terdapat $-u \in V$ sedemikian sehingga harus memenuhi
$$u + (-u) = (-u) + u = 0$$
6) **Distributif**: Untuk semua $u, v, w \in V$ dan k, m adalah skalar, maka harus memenuhi
$$k(u + v) = ku + kv$$
$$(k + m)w = kw + mw$$
$$k(mu) = (km)u$$

C. Teori Grup (Group Theory)

Sebuah Grup $(G, *)$ terdiri dari himpunan tidak kosong G dan sebuah operasi biner $*$ yang memetakan dua elemen G menjadi elemen lain di G. Struktur sebuah Grup harus memenuhi empat aksioma dasar berikut, yaitu:

1) **Tertutup (Closure)**: Untuk setiap $a, b \in G$, hasil operasi $a \cdot b$ juga harus menghasilkan sebuah elemen yang merupakan anggota dari G.
2) **Asosiatif**: Untuk setiap $a, b, c \in G$, berlaku $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
3) **Identitas**: Terdapat satu elemen unik $x \in G$ sedemikian sehingga untuk setiap $a \in G$, berlaku $a \cdot x = x \cdot a = a$.
4) **Invers**: Untuk setiap anggota $a \in G$, terdapat anggota $a^{-1} \in G$ sedemikian sehingga $a \cdot a^{-1} = a^{-1} \cdot a = x$.

Jika sebuah Grup memenuhi satu aksioma tambahan yaitu: **Komutatif**, di mana untuk setiap $a, b \in G$ memenuhi $a \cdot b = b \cdot a$. Maka Grup tersebut disebut sebagai **Grup Abelian**.

D. Lapangan Hingga (Galois Field)

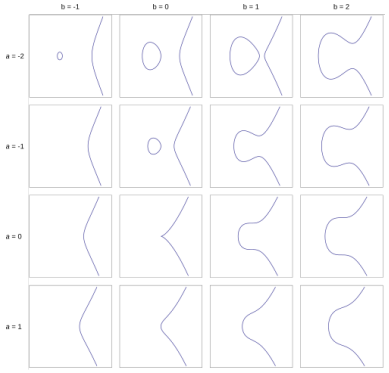
Lapangan hingga adalah sebuah himpunan yang memiliki jumlah elemen terbatas, di mana operasi penjumlahan, pengurangan, perkalian, dan pembagian (kecuali pembagian dengan nol) terdefinisi dan memenuhi aksioma: asosiatif, komutatif, distributif, dan invers.

Jumlah elemen q dalam lapangan disebut sebagai orde dari lapangan. Contohnya dalam penerapan ECC dasar, umum digunakan sebuah lapangan prima $(\mathbb{F}_p)$, di mana p adalah sebuah bilangan prima yang sangat besar.

Operasi aritmatika dalam $\mathbb{F}_p$ dilakukan dengan menggunakan konsep aritmatika modular. Untuk setiap bilangan bulat a dan b, maka:

- **Penjumlahan**: $(a + b)$ mod p
- **Perkalian**: $(a \cdot b)$ mod p
- **Invers Penjumlahan (Negatif)**: $-a$ adalah sebuah bilangan x sedemikian sehingga memenuhi $a + x \equiv 0$ mod p
- **Invers Perkalian (Pembagian)**: a^{-1} adalah bilangan x sedemikian sehingga $a \cdot x \equiv 1$ mod p

E. Kurva Eliptik



Gambar 1. Kurva eliptik

Sebuah Kurva Eliptik adalah kurva dengan bentuk umum persamaan sebagai berikut:

$$y^2 = x^3 + ax + b$$

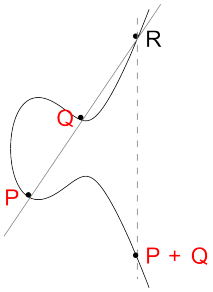
dengan syarat bahwa $4a^3 + 27b^2 \neq 0$. Tiap nilai a dan b yang berbeda menghasilkan sebuah kurva eliptik yang berbeda juga.

Kurva eliptik terdefinisi untuk setiap $x, y \in \mathbb{R}$. Di dalam kurva eliptik ini terdapat sebuah titik O(x, ∞), yaitu titik pada infinity. Titik-titik P(x, y) pada kurva eliptik bersama dengan operasi + membentuk sebuah grup abelian.

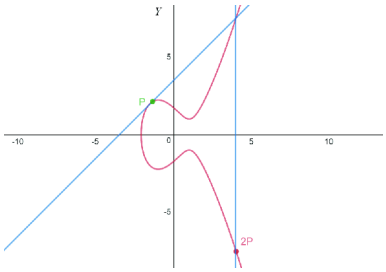
Misal, terdapat sebuah titik $P, Q,$ dan R yang merupakan titik-titik pada kurva eliptik dan $P \neq Q$, maka operasi $P + Q = R$ memiliki arti:

1) Menarik garis yang melalui P dan Q
2) Garis tersebut akan memotong kurva pada titik$-R$
3) Pencerminan titik $-R$ terhadap sumbu-x adalah titik R dengan keterangan jika $R = (x, y)$ maka $-R$ adalah titik $(x, -y)$
4) Titik R adalah hasil dari penjumlahan titik P dan Q

Sedangkan, apabila terjadi penjumlahan titik pada kurva eliptik dengan $P = Q$, maka $P + Q$ ekivalen dengan $2P$. Operasi penjumlahan titik ini akan menghasilkan sebuah titik R yang didapat dari perpotongan garis singgung pada titik P dengan garis kurva eliptik.



Gambar 2. Penjumlahan titik pada kurva eliptik



Gambar 3. Perkalian titik dengan skalar pada kurva eliptik

F. Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) adalah kriptografi dengan jenis sistem kriptografi asimetrik dengan melibatkan kurva eliptik. ECC dikembangkan oleh Neal Koblitz dan Victor S. Miller pada tahun 1985. Dibandingkan dengan sistem kriptografi lainnya seperti RSA, ECC bisa memiliki ukuran kunci yang lebih kecil dengan tingkat keamanan yang sama.

Mekanisme pembangkitan kunci dari ECC sendiri menggunakan Kurva Eliptik di atas sebuah Lapangan Hingga $\mathbb{F}_p$. Perubahan kurva eliptik ke dalam lapangan hingga bertujuan untuk mengubah range yang merupakan himpunan bilangan real tak hingga menjadi sebuah range bilangan bulat berhingga sehingga memudahkan kalkulasi enkripsi. Mekanisme lebih jelasnya dari pembangkitan kedua kunci ini adalah sebagai berikut:

- **Kunci Privat (Private Key)**: Sebuah bilangan bulat acak d (skalar) yang dipilih dari rentang 1 hingga n-1, di mana n adalah orde dari titik generator.
- **Kunci Public (Public Key)**: Sebuah titik Q pada kurva yang merupakan hasil dari operasi perkalian skalar titik generator G dengan kunci privat d.

$$Q = d \cdot G = \underbrace{G + G + ... + G}_{\text{d kali}}$$

Keamanan enkripsi dari ECC terletak pada masalah logaritma diskrit dengan nama *Elliptic Curve Discrete Logarithm Problem* (ECDLP). Ini disebabkan karena berat dan lamanya waktu komputasi untuk pembalikan operasi skalar. Lebih jelasnya dari masalah ini adalah sebagai berikut:

"Diberikan sebuah kurva eliptik $E(\mathbb{F}_p)$, sebuah titik dasar G, dan titik hasil $Q = d \cdot G$. Sangatlah mudah untuk menghitung Q jika diketahui nilai d. Namun, sangat sulit secara komputasi untuk menghasilkan nilai skalar d jika hanya diketahui titik G dan Q."

III. Analisis Perhitungan

Sebelum melakukan pengujian langsung pada kode, perlu diperjelas terlebih dahulu mengenai perbedaan kedua pendekatan dalam perhitungan perkalian point dengan skalar pada kurva eliptik. Pada bagian ini akan dilakukan penurunan rumus dan pengerjaan langkah demi langkah dari masing-masing pendekatan.

A. Pendekatan dengan Koordinat Affine

Diberikan sebuah persamaan yang menggambarkan kurva eliptik E di atas sebuah lapangan hingga $\mathbb{F}_p$ sebagai berikut:

$$y^2 = x^3 + ax + b \pmod{p}$$

Misalkan terdapat sebuah titik $P = (x_1, y_1)$ dalam kurva eliptik E dan k adalah sebuah skalar, akan dicari titik hasil penggandaan $kP = (x_3, y_3)$ yang merupakan titik dalam kurva eliptik E juga. Tahapan dalam mencari hasil penggandaan ini adalah sebagai berikut:

1) **Penggandaan Titik**, untuk kasus apabila terdapat n untuk $k = 2^n$ di mana n bilangan asli
   a) **Mencari gradien garis singgung** λ
   Misal λ melambangkan turunan implisit dari persamaan kurva, maka λ diperoleh dengan cara mencari turunan pertama dari persamaan awal kurva eliptik pada lapangan hingga. Persamaan awal kurva eliptik standar dalam bilangan riil adalah sebagai berikut:

   $$y^2 = x^3 + ax + b$$

   Turunan pertama persamaan di atas menjadi:

   $$\frac{d}{dx}(y^2) = \frac{d}{dx}(x^3 + ax + b)$$

   menerapkan aturan rantai pada ruas kiri menjadi

   $$2y \cdot \tfrac{dy}{dx}$$

   $$2y\frac{dy}{dx} = 3x^2 + a$$

   $$\frac{dy}{dx} = \frac{3x^2 + a}{2y}$$

   Didapatkan bahwa λ adalah gradien perubahan y terhadap perubahan x dari kurva eliptik pada lapangan hingga di titik $P(x_1, y_1)$. Sehingga didapatkan bahwa λ adalah

   $$\lambda \equiv \frac{3x_1^2 + a}{2y} \pmod{p}$$

   Penghitungan invers modular $(2y_1)^{-1} \pmod{p}$ akan menghabiskan waktu yang cukup lama.
   b) **Menghitung Absis Baru** (x_3)
   Diketahui bahwa persamaan garis singgung adalah $y = \lambda(x - x_1) + y_1$. Persamaan y ini dapat disubtitusi ke dalam persamaan kurva dan menghasilkan sebuah polinomial berderajat 3. Berdasarkan Teorema Vieta mengenai hubungan

suatu akar-akar polinomial, jumlah ketiga akar (x_1, x_2, x_3) sama dengan kuadrat dari gradien, sehingga dapat diturunkan persamaan untuk mencari x_3 sebagai berikut:

$$x_1 + x_1 + x_3 = \lambda^2 \pmod p$$
$$2x_1 + x_3 = \lambda^2 \pmod p$$
$$x_3 = \lambda^2 - 2x_1 \pmod p$$

c) **Menghitung Ordinat Baru** (y_3)
Nilai y pada garis dari posisi x_3 adalah $y' = \lambda(x_3 - x_1) + y_1$. Titik hasil penjumlahan adalah pencerminan y' terhadap sumbu-x sehingga:

$$y_3 = -(\lambda(x_3 - x_1) + y_1) \pmod p$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$$

2) **Penjumlahan Titik**, untuk kasus apabila k tidak memenuhi $k = 2^n$ untuk setiap bilangan asli n, maka diperlukan tahap penjumlahan dua titik yang berbeda pada kurva eliptik.

a) **Menghitung Gradien Garis yang Melalui Dua Titik** λ
Misal, didefinisikan dua titik berbeda yaitu $P = (x_1, y_1)$ dan $Q = (x_2, y_2)$ pada kurva eliptik, maka gradien untuk garis yang melalui dua titik didefinisikan sebagai:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod p$$

Penghitungan invers modular $(x_2 - x_1)^{-1} \pmod p$ akan menghabiskan waktu yang cukup lama.

b) **Menghitung Absis Baru** (x_3)
Menggunakan prinsip sama dengan penggandaan, tetapi berbeda pada akarnya, di mana $x_1 \neq x_2$. Maka dapat dicari nilai x_3 sebagai berikut:

$$x_1 + x_2 + x_3 = \lambda^2 \pmod p$$
$$x_3 = \lambda^2 - x_1 - x_2 \pmod p$$

c) **Menghitung Ordinat Baru** (y_3)
Sama dengan rumus pengulangan, di mana y_3 adalah

$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$$

B. Pendekatan dengan Proyeksi Koordinat Berbobot

Pendekatan proyeksi koordinat memandang titik dalam kurva eliptik E sebagai objek yang ekuivalen dalam ruang proyektif berbobot $\mathbb{P}(2, 3, 1)$. Berbeda dengan ruang proyektif standar, ruang proyektif berbobot memberikan skala yang berbeda tiap masing-masing komponen. Lebih jelasnya bobot dari masing-masing komponen adalah sebagai berikut:

- Koordinat X diberi bobot 2
- Koordinat Y diberi bobot 3
- Koordinat Z diberi bobot 1

Artinya, sebuah titik Affine (x, y) dapat direpresentasikan oleh vektor (X, Y, Z) dengan hubungan pemetaan sebagai berikut:

$$x = \frac{X}{Z^2},\ y = \frac{Y}{Z^3}$$

Pembobotan $\mathbb{P}(2, 3, 1)$ dipilih untuk mengubah persamaan kurva eliptik menjadi homogen atau seimbang derajat pangkatnya. Persamaan awal:

$$y^2 = x^3 + ax + b$$

- Ruas kiri (y^2) memiliki derajat pangkat 2
- Ruas kanan (x^3) memiliki derajat pangkat 3

Perbedaan derajat pangkat membuat komputasi aljabar menjadi lama. Sehingga, dibentuk sebuah bobot yang menyeimbangkan total derajat kiri dengan total derajat kanan. Diketahui KPK dari 2 dan 3 adalah 6, maka:

- Jika y berbobot 3, maka y^2 memiliki "besar" $3 \times 2 = 6$
- Jika x berbobot 2, maka x^3 memiliki "besar" $2 \times 3 = 6$

Subtitusi pemetaan koordinat proyektif berbobot ke dalam kurva akan membuktikan hal ini:

$$\left(\frac{Y}{Z^3}\right)^2 = \left(\frac{X}{Z^2}\right)^3 + a\left(\frac{X}{Z^2}\right) + b$$
$$\frac{Y^2}{Z^6} = \frac{X^3}{Z^6} + \frac{aX}{Z^2} + b$$

Mengalikan kedua ruas dengan Z^6:

$$Y^2 = X^3 + aXZ^4 + bZ^4$$

Persamaan akhir membentuk sebuah persamaan tanpa bentuk pecahan dengan pengerjaan operasi yang lebih efisien nantinya.

Selanjutnya, akan ditulis langkah matematis dari implementasi operasi perkalian dengan skalar dan penjumlahan titik. Misalkan terdapat sebuah titik $P(X_1, Y_1, Z_2)$ dalam koordinat proyektif berbobot dan skalar k, akan dicari nilai dari $kP = (X_3, Y_3, Z_3)$.

1) **Penggandaan Titik**, untuk kasus apabila terdapat n untuk $k = 2^n$ di mana n adalah bilangan asli

a) **Pencarian Gradien** λ
Didefinisikan rumus dasar untuk mencari gradien garis singgung pada koordinat Affine sebagai berikut:

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

Dengan melakukan subtitusi koordinat proyektif berbobot $x_1 = \frac{X_1}{Z_1^2}$ dan $y_1 = \frac{Y_1}{Z_1^3}$ ke dalam rumus akan didapat:

$$\lambda = \frac{3\left(\frac{X_1}{Z_1^2}\right)^2 + a}{2\left(\frac{Y_1}{Z_1^3}\right)} \pmod p = \frac{\frac{3X_1^2}{Z_1^4} + a}{\frac{2Y_1}{Z_1^3}} \pmod p$$
$$\lambda = \frac{\frac{3X_1^2 + aZ_1^4}{Z_1^4}}{\frac{2Y_1}{Z_1^3}} \pmod p$$
$$\lambda = \frac{3X_1^2 + aZ_1^4}{Z_1^4} \cdot \frac{Z_1^3}{2Y_1} \pmod p$$
$$\lambda = \frac{3X_1^2 + aZ_1^4}{2Y_1Z_1} \pmod p$$

Di sini didapatkan sebuah komponen vektor gradien

- Pembilang (M): $3X_1^2 + aZ_1^4$
- Penyebut (Z_3): $2Y_1Z_1$

Maka gradien bisa ditulis menjadi $\lambda = \frac{M}{Z_3}$ tanpa perlu menghitung hasil modulo nya terlebih dahulu.

b) **Pencarian Absis Baru** (X_3)
Diketahui rumus Affine untuk x_3 adalah:

$$x_3 = \lambda^2 - 2x_1$$

Maka dapat disubtitusikan pencarian X_3 dengan memasukkan $x_3 = \frac{X_3}{Z_3^2}$ dan λ ke dalam rumus seperti berikut:

$$\frac{X_3}{Z_3^2} = \left(\frac{M}{Z_3}\right)^2 - 2\left(\frac{X_1}{Z_1^2}\right) \pmod p$$
$$\frac{X_3}{Z_3^2} = \frac{M^2}{Z_3^2} - \frac{2X_1}{Z_1^2} \pmod p$$

Karena diketahui $Z_3 = 2Y_1Z_1$, maka $Z_3^2 = 4Y_1^2Z_1^2$. Sehingga agar penyebut suku kedua berubah menjadi Z_3^2, kalikan penyebut dan pembilang suku kedua dengan $4Y_1^2$

$$\frac{X_3}{Z_3^2} = \frac{M^2}{Z_3^2} - \frac{2X_1}{Z_1^2} \cdot \frac{(4Y_1^2)}{(4Y_1^2)} \pmod p$$
$$\frac{X_3}{Z_3^2} = \frac{M^2}{Z_3^2} - \frac{8X_1Y_1^2}{Z_3^2} \pmod p$$

Sehingga X_3 adalah

$$X_3 = M^2 - 8X_1Y_1^2 \pmod p$$

c) **Pencarian Ordinat Baru** (Y_3)
Diketahui rumus untuk mencari y_3 pada Affine adalah:

$$y_3 = \lambda(x_1 - x_3) - y_1$$

Subtitusi proyeksi koordinat berbobot ke dalam persamaan untuk mencari Y_3

$$\frac{Y_3}{X_3^3} = \frac{M}{Z_3}\left(\frac{S - X_3}{Z_3^3}\right) - \frac{Y_1}{Z_1^3} \pmod p$$
$$\frac{Y_3}{Z_3^3} = \frac{M(S - X_3)}{Z_3^3} - \frac{Y_1}{Z_1^3} \pmod p$$

Mengubah suku terakhir dengan cara mengalikan pembilang dan penyebutnya dengan $8Y_1^3$ sehingga penyebutnya akan membentuk $Z_3^3 = 8Y_1^3Z_1^3$

$$\frac{Y_3}{X_3^3} = \frac{M(S - X_3)}{Z_3^3} - \frac{Y_1 \cdot (8Y_1^3)}{Z_1^3 \cdot (8Y_1^3)} \pmod p$$
$$\frac{Y_3}{X_3^3} = \frac{M(S - X_3) - 8Y_1^4}{Z_3^3} \pmod p$$

Sehingga didapatkan bahwa nilai Y_3 adalah:

$$Y_3 = M(S - X_3) - 8Y_1^4 \pmod p$$

2) **Penjumlahan Titik**, untuk kasus apabila tidak ada n yang memenuhi $k = 2^n$ untuk seluruh n bilangan asli

a) **Normalisasi Silang (Cross-Normalization)**
Perlu membandingkan nilai absis x_1 dan x_2. Dalam koordinat proyeksi berbobot, definisinya adalah:

$$x_1 = \frac{X_1}{Z_1^2}, \quad x_2 = \frac{X_2}{Z_2^2}$$

Untuk dapat mengoperasikan keduanya, samakan penyebutnya menjadi $Z_1^2Z_2^2$.

- Untuk x_1, kalikan pembilang dan penyebut dengan Z_2^2:

$$x_1 = \frac{X_1Z_2^2}{Z_1^2Z_2^2}$$

Kita definisikan variabel baru U_1 untuk pembilang:

$$U_1 = vX_1Z_2^2 \pmod p$$

- Untuk x_2, kalikan pembilang dan penyebut dengan Z_1^2:

$$x_2 = \frac{X_2Z_1^2}{Z_1^2Z_2^2}$$

Kita definisikan variabel baru U_2 untuk pembilang:

$$U_2 = X_2Z_1^2 \pmod p$$

Selanjutnya, lakukan hal yang sama untuk ordinat y dengan penyebut bersama $Z_1^3Z_2^3$.

- Untuk y_1:

$$y_1 = \frac{Y_1}{Z_1^3} = \frac{Y_1Z_2^3}{Z_1^3Z_2^3}$$

Kita definisikan variabel baru S_1:

$$S_1 = Y_1Z_2^3 \pmod p \tag{1}$$

- Untuk y_2:

$$y_2 = \frac{Y_2}{Z_2^3} = \frac{Y_2Z_1^3}{Z_1^3Z_2^3}$$

Kita definisikan variabel baru S_2:

$$S_2 = Y_2Z_1^3 \pmod p \tag{2}$$

b) **Analisis Gradien** λ Rumus gradien garis antara dua titik pada koordinat Affine adalah:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod p$$

Substitusikan nilai-nilai vektor yang sudah dinormalisasi di atas:

$$\lambda = \frac{\frac{S_2 - S_1}{(Z_1Z_2)^3}}{\frac{U_2 - U_1}{(Z_1Z_2)^2}} \pmod p$$

Lakukan operasi pembagian pecahan:

$$\lambda = \frac{S_2 - S_1}{(Z_1Z_2)^3} \cdot \frac{(Z_1Z_2)^2}{U_2 - U_1} \pmod p$$

Sederhanakan variabel Z. Suku $(Z_1Z_2)^2$ di pembilang mencoret sebagian dari pangkat 3 di penyebut, menyisakan satu faktor (Z_1Z_2) di penyebut:

$$\lambda = \frac{S_2 - S_1}{(U_2 - U_1) \cdot Z_1Z_2} \pmod p$$

Untuk menyederhanakan notasi, kita definisikan selisih komponen vektor sebagai berikut:

$R = S_2 - S_1 \pmod p$ (Selisih Ordinat Proyektif)
$H = U_2 - U_1 \pmod p$ (Selisih Absis Proyektif)

Maka persamaan gradien menjadi:

$$\lambda = \frac{R}{H \cdot Z_1Z_2}$$

c) **Definisi Komponen** Z_3
Penyebut dari gradien di atas adalah $H \cdot Z_1Z_2$. Dapat ditetapkan nilai ini sebagai komponen Z untuk vektor hasil (Z_3):

$$Z_3 = H \cdot Z_1 \cdot Z_2 \pmod p$$

d) **Pencarian Komponen** X_3
Rumus Affine untuk absis baru adalah:

$$x_3 = \lambda^2 - x_1 - x_2 \pmod p$$

Target kita adalah mendapatkan bentuk vektor $\frac{X_3}{Z_3^2}$. Perhatikan bahwa karena $Z_3 = HZ_1Z_2$, maka $Z_3^2 = H^2(Z_1Z_2)^2$. Substitusikan λ dan nilai x yang telah dinormalisasi:

$$\frac{X_3}{Z_3^2} = \left(\frac{R}{HZ_1Z_2}\right)^2 - \frac{U_1}{(Z_1Z_2)^2} - \frac{U_2}{(Z_1Z_2)^2} \pmod p$$

$$\frac{X_3}{Z_3^2} = \frac{R^2}{H^2(Z_1Z_2)^2} - \frac{U_1 + U_2}{(Z_1Z_2)^2} \pmod p$$

Suku kedua dan ketiga memiliki penyebut $(Z_1Z_2)^2$. Agar penyebutnya menjadi sama dengan suku pertama (yaitu Z_3^2), kalikan pembilang dan penyebut suku tersebut dengan H^2:

$$\frac{X_3}{Z_3^2} = \frac{R^2}{Z_3^2} - \frac{(U_1 + U_2)H^2}{H^2(Z_1Z_2)^2} \pmod p$$

Karena seluruh penyebut sudah seragam (Z_3^2), dapat diambil pembilangnya saja sebagai hasil akhir komponen X_3:

$$X_3 = R^2 - H^2(U_1 + U_2) \pmod p$$

e) **Pencarian Komponen** Y_3
Rumus Affine untuk ordinat baru adalah:

$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$$

Target kita adalah mendapatkan bentuk vektor $\frac{Y_3}{Z_3^3}$. Lakukan substitusi proporsional:

$$\frac{Y_3}{Z_3^3} = \frac{R}{HZ_1Z_2}\left(\frac{U_1H^2}{Z_3^2} - \frac{X_3}{Z_3^2}\right) - \frac{S_1}{(Z_1Z_2)^3} \pmod p$$

Perhatikan suku pertama (gradien dikali selisih x). Penyebutnya menjadi $(HZ_1Z_2) \cdot Z_3^2 = Z_3^3$, yang sudah sesuai target. Namun, suku terakhir (S_1) masih memiliki penyebut $(Z_1Z_2)^3$. Agar penyebutnya menjadi $Z_3^3 = H^3(Z_1Z_2)^3$, kalikan pembilang dan penyebutnya dengan H^3:

$$\frac{Y_3}{Z_3^3} = \frac{R(U_1H^2 - X_3)}{Z_3^3} - \frac{S_1H^3}{H^3(Z_1Z_2)^3} \pmod p$$

Karena penyebut sudah sama (Z_3^3), ambil pembilangnya sebagai hasil akhir komponen Y_3:

$$Y_3 = R(U_1H^2 - X_3) - S_1H^3 \pmod p$$

IV. Percobaan

Implementasi dalam bentuk kode dan pengujian akan dilakukan dengan menggunakan bahasa pemrograman C. Pengujian dilakukan dengan cara membuat masing-masing fungsi dari masing-masing pendekatan. Setelah itu, waktu akan dihitung menggunakan library *time.h* pada C. Waktu dimulai pada saat sebelum kalkulasi pertama pada perkalian titik dengan skalar k. Implementasi dari kode C ini adalah sebagai berikut.

A. Implementasi Kode

```c
// === DEKLARASI STRUKTUR DAN VARIABEL GLOBAL === //

// Deklarasi variabel global untuk parameter kurva eliptik
int64_t p;
int64_t a;
int64_t b;

// Deginisi struktur titik dalam koordinat affine
typedef struct {
    int64_t x;
    int64_t y;
    int is_inf;
} AffinePoint;

// Deginisi struktur titik dalam koordinat proyeksi berbobot
typedef struct {
    int64_t X;
    int64_t Y;
    int64_t Z;
    int is_inf;
} WeightedPoint;

// === KONVERSI KOORDINAT === /

// Konversi antara koordinat affine dan proyeksi berbobot
WeightedPoint toWeighted(AffinePoint P) {
    return (WeightedPoint){P.x, P.y, 1, P.is_inf};
}

// Konversi antara koordinat proyeksi berbobot dan affine
AffinePoint toAffine(WeightedPoint P) {
    if (P.is_inf || P.Z == 0) return (AffinePoint){0, 0, 1};

    int64_t Z_inv = modInv(P.Z);
    int64_t Z_inv_sq = modSqr(Z_inv);
    int64_t Z_inv_cu = modMul(Z_inv, Z_inv_sq);

    int64_t x = modMul(P.X, Z_inv_sq);
    int64_t y = modMul(P.Y, Z_inv_cu);

    return (AffinePoint){x, y, 0};
}

// === METODE AFFINE === //

// Operasi perkalian double titik pada kurva eliptik
AffinePoint affineDouble(AffinePoint P) {
    if (P.is_inf) return P;
    int64_t num = modAdd(modMul(3, modSqr(P.x)), a);
    int64_t den = modMul(2, P.y);

    if (den == 0) return (AffinePoint){0, 0, 1};
    int64_t inv = modInv(den);
    int64_t lambda = modMul(num, inv);
    int64_t x3 = modSub(modSqr(lambda), modMul(2, P.x));
    int64_t y3 = modSub(modMul(lambda, modSub(P.x, x3)), P.y);
    AffinePoint R = {x3, y3, 0};
    return R;
}

// Operasi penjumlahan titik pada kurva eliptik
AffinePoint affineAdd(AffinePoint P, AffinePoint Q) {
    if (P.is_inf) return Q;
    if (Q.is_inf) return P;
```

```
    if (P.x == Q.x && P.y == Q.y) return affineDouble(P);
    int64_t num = modSub(Q.y, P.y);
    int64_t den = modSub(Q.x, P.x);

    if (den == 0) return (AffinePoint){0, 0, 1};
    int64_t inv = modInv(den);
    int64_t lambda = modMul(num, inv);
    int64_t x3 = modSub(modSub(modSqr(lambda), P.x), Q.x);
    int64_t y3 = modSub(modMul(lambda, modSub(P.x, x3)), P.y);
    AffinePoint R = {x3, y3, 0};
    return R;
}

// Operasi perkalian skalar titik pada kurva eliptik
AffinePoint affineScalarMul(AffinePoint P, int64_t k) {
    AffinePoint R = {0, 0, 1};
    AffinePoint Temp = P;
    while (k > 0) {
        if (k % 2 == 1) R = affineAdd(R, Temp);
        Temp = affineDouble(Temp);
        k /= 2;
    }
    return R;
}

AffinePoint affineScalarMul(AffinePoint P, int64_t k) {
    AffinePoint R = {0, 0, 1}; // Titik Infinity
    AffinePoint Temp = P;
    while (k > 0) {
        if (k % 2 == 1) R = affineAdd(R, Temp);
        Temp = affineDouble(Temp);
        k /= 2;
    }
    return R;
}

// === METODE PROYEKSI BERBOBOT === //

// Operasi perkalian double titik pada kurva eliptik dalam proyeksi berbobot
WeightedPoint weightedDouble(WeightedPoint P) {
    if (P.is_inf) return P;

    int64_t X1 = P.X; int64_t Y1 = P.Y; int64_t Z1 = P.Z;
    int64_t A = modSqr(X1);
    int64_t B = modSqr(Y1);
    int64_t C = modSqr(B);
    int64_t Z1_sq = modSqr(Z1);
    int64_t Z1_4 = modSqr(Z1_sq);
    int64_t M = modAdd(modMul(3, A), modMul(a, Z1_4));
    int64_t Z3 = modMul(2, modMul(Y1, Z1));
    int64_t S = modMul(4, modMul(X1, B));
    int64_t X3 = modSub(modSqr(M), modMul(2, S));
    int64_t Y3 = modSub(modMul(M, modSub(S, X3)), modMul(8, C));
    WeightedPoint R = {X3, Y3, Z3, 0};
    return R;
}

// Operasi penjumlahan titik pada kurva eliptik dalam proyeksi berbobot
WeightedPoint weightedAdd(WeightedPoint P, WeightedPoint Q) {
    if (P.is_inf) return Q;
    if (Q.is_inf) return P;
    int64_t Z1_sq = modSqr(P.Z);
    int64_t Z2_sq = modSqr(Q.Z);
    int64_t U1 = modMul(P.X, Z2_sq);
    int64_t U2 = modMul(Q.X, Z1_sq);
    int64_t S1 = modMul(P.Y, modMul(Q.Z, Z2_sq));
    int64_t S2 = modMul(Q.Y, modMul(P.Z, Z1_sq));

    if (U1 == U2) {
        if (S1 != S2) return (WeightedPoint){0,0,0,1};
        return weightedDouble(P);
    }

    int64_t H = modSub(U2, U1);
    int64_t R = modSub(S2, S1);
    int64_t Z3 = modMul(modMul(P.Z, Q.Z), H);
    int64_t H_sq = modSqr(H);
    int64_t H_cu = modMul(H, H_sq);
    int64_t term = modMul(2, modMul(U1, H_sq));
    int64_t X3 = modSub(modSub(modSqr(R), H_cu), term);
    int64_t Y3 = modSub(modMul(R, modSub(modMul(U1, H_sq), X3)), modMul(S1, H_cu)
          );
    WeightedPoint Res = {X3, Y3, Z3, 0};
    return Res;
}

// Operasi perkalian skalar titik pada kurva eliptik dengan
WeightedPoint weightedScalarMul(WeightedPoint P, int64_t k) {
    WeightedPoint R = {0, 1, 0, 1};
    WeightedPoint Temp = P;
    while (k > 0) {
        if (k % 2 == 1) R = weightedAdd(R, Temp);
        Temp = weightedDouble(Temp);
        k /= 2;
    }
    return R;
}
```

Listing 1: Implementasi Operasi Inti ECC

Untuk kode lengkapnya, bisa dilihat pada link GitHub yang berada di lampiran.

B. Hasil Pengujian Program

Dipilih beberapa test-case yang digunakan untuk menguji kedua pendekatan. Dipilih 3 test-case dengan tiga tingkatan,

```
Masukkan Bilangan Prima (p): 17
Masukkan Parameter Kurva a: 2
Masukkan Parameter Kurva b: 2
Masukkan Titik Generator x: 5
Masukkan Titik Generator y: 1
Masukkan Nilai Skalar k: 7

Melakukan benchmark 2000 iterasi...

=== HASIL PERHITUNGAN ===
Metode Affine             : (0, 6)
Metode Proyeksi Berbobot  : (0, 6)

=== PERBANDINGAN WAKTU ===
Waktu Affine              : 0.005000 detik
Waktu Proyeksi Berbobot   : 0.006000 detik
Peningkatan Kecepatan     : 0.83x
```

Gambar 4. Hasil Kasus Uji 1 (Ringan)

```
Masukkan Bilangan Prima (p): 23
Masukkan Parameter Kurva a: 1
Masukkan Parameter Kurva b: 1
Masukkan Titik Generator x: 3
Masukkan Titik Generator y: 10
Masukkan Nilai Skalar k: 9

Melakukan benchmark 2000 iterasi...

=== HASIL PERHITUNGAN ===
Metode Affine             : (0, 1)
Metode Proyeksi Berbobot  : (0, 1)

=== PERBANDINGAN WAKTU ===
Waktu Affine              : 0.006000 detik
Waktu Proyeksi Berbobot   : 0.007000 detik
Peningkatan Kecepatan     : 0.86x
```

Gambar 5. Hasil Kasus Uji 2 (Sedang)

```
Masukkan Bilangan Prima (p): 100003
Masukkan Parameter Kurva a: 1
Masukkan Parameter Kurva b: 1
Masukkan Titik Generator x: 0
Masukkan Titik Generator y: 1
Masukkan Nilai Skalar k: 12345

Melakukan benchmark 2000 iterasi...

=== HASIL PERHITUNGAN ===
Metode Affine             : (48282, 75415)
Metode Proyeksi Berbobot  : (48282, 75415)

=== PERBANDINGAN WAKTU ===
Waktu Affine              : 0.047000 detik
Waktu Proyeksi Berbobot   : 0.024000 detik
Peningkatan Kecepatan     : 1.96x
```

Gambar 6. Hasil Kasus Uji 3 (Sulit)

yakni pemrosesan yang rendah, sedang, dan tinggi. Berikut adalah hasil dari pengujiannya:

V. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, didapatkan dari tiga kasus uji bahwa pendekatan koordinat proyeksi berbobot memiliki kecepatan kalkulasi hampir **dua kali lipat** lebih cepat dari metode koordinat Affine biasa untuk perhitungan yang besar. Ini disebabkan karena perhitungan invers modular pada pendekatan proyeksi yang dilakukan di akhir. Tetapi, dengan kasus uji yang kecil waktu kecepatan pendekatan proyeksi memiliki kecepatan

sama atau bahkan lebih lama sedikit dibanding pendekatan Affine. Hal ini dikarenakan overhead yang dimiliki dari pendekatan proyeksi, karena harus melakukan transformasi terlebih dahulu. Walaupun begitu, pendekatan proyeksi lebih baik dibandingkan pendekatan Affine. Hal ini dikarenakan di dunia nyata kunci enkripsi berukuran sangat besar, sehingga pendekatan proyeksi lebih cocok untuk kalkulasi ini.

VI. Ucapan Terima Kasih

Pertama-tama penulis ingin mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas anugerah yang telah diberikan sehingga penulis bisa menyelesaikan makalah ini. Penulis juga ingin berterima kasih kepada dosen yang mengajar penulis pada mata kuliah IF2123 Aljabar Linear & Geometri, yaitu bapak Ir. Rila Mandala, M.Eng., Ph.D. Selanjutnya penulis juga ingin memberikan ucapan terima kasih kepada keluarga dan teman-teman yang selalu memberikan dukungan selama perkuliahan di Semester 3 ini.

Referensi

[1] D. Hankerson, A. Menezes, dan S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York: Springer, 2004.

[2] Penulis D. Hegade, "The Role of Algebraic Geometry In Cryptography," *International Journal of Research and Analytical Reviews (IJRAR)*, Paper ID IJRAR23B4687. [File: IJRAR23B4687.pdf].

[3] R. Mandala, "Vektor di Ruang Euclidean Bagian 1," Slide Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika, Institut Teknologi Bandung, 2025. [File: Algeo-11.pdf].

[4] R. Mandala, "Vektor di Ruang Euclidean Bagian 2," Slide Kuliah IF2123 Aljabar Linier dan Geometri, Program Studi Teknik Informatika, Institut Teknologi Bandung, 2025. [File: Algeo-12.pdf].

[5] J.S. Milne, *Fields and Galois Theory*. 2022. Available: https://www.jmilne.org/math/CourseNotes/FT.pdf

[6] J.S. Milne, *Group Theory*. 2025. Available: https://www.jmilne.org/math/CourseNotes/GT.pdf

[7] R. Munir, "Elliptic Curve Cryptography (ECC)," Materi Referensi, 2013. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/ECC%20(2013).pdf

[8] GeeksforGeeks, "Blockchain - Elliptic Curve Cryptography," *GeeksforGeeks*, [Online]. Available: https://www.geeksforgeeks.org/ethical-hacking/blockchain-elliptic-curve-cryptography/. [Diakses: 23-Des-2025].

[9] redshiftzero, "Elliptic Curve Cryptography 101," *YouTube*, [Video]. Available: https://youtu.be/Xem-AjUBOkU?si=mZ-DKGug_MF_CwM9. [Diakses: 23-Des-2025].

Lampiran GitHub dan Video

1) https://github.com/renuno-frinardi/IF2123-Makalah-AljabarLinear-Geometri.git
2) https://youtu.be/IGv2PYzB3Qo

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025

Renuno Yuqa Frinardi, 13524080