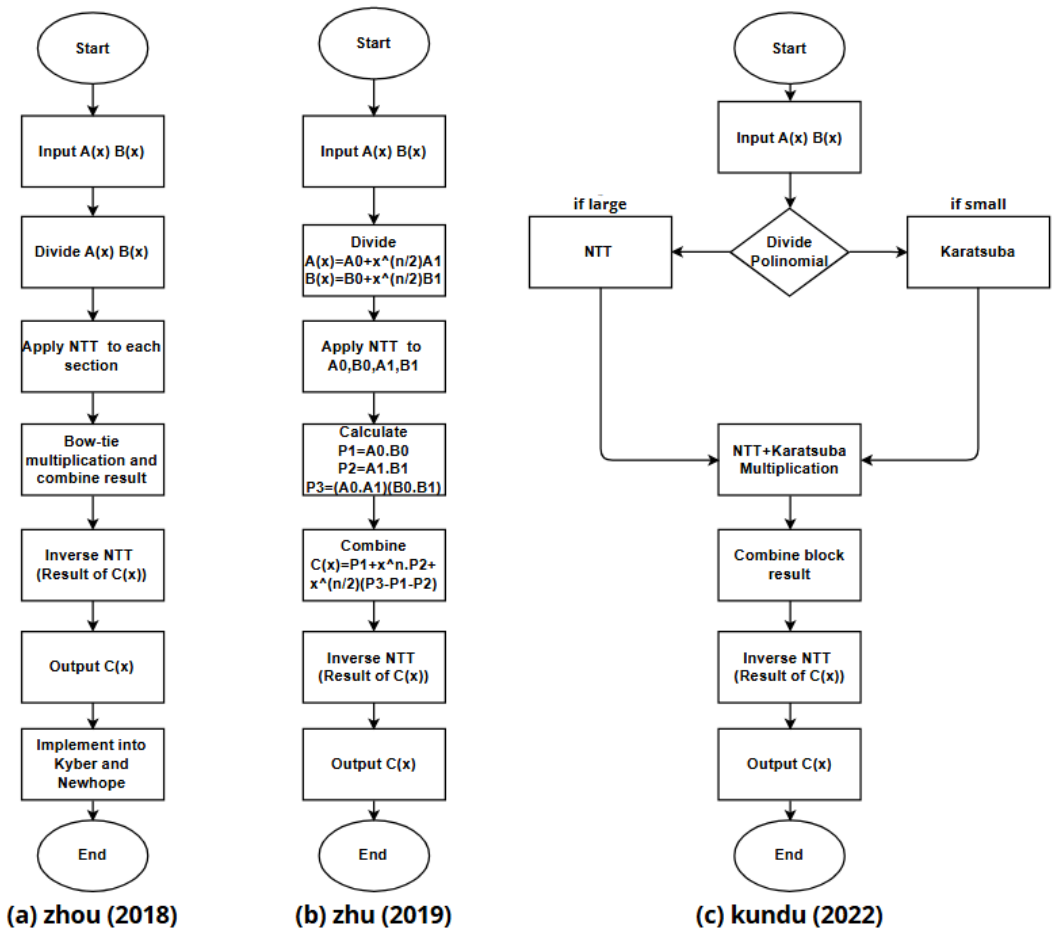
BAB III

METODOLOGI PENELITIAN

3.1 Metodologi

Algoritma NTRU bekerja secara matematis berbasis polinomial yang dihitung pada ring $\mathrm{Z}_q[x]/(x^n - 1)$. Artinya, operasi enkripsi maupun dekripsi dilakukan melalui perhitungan polinomial dengan derajat tertentu, di mana setiap koefisien diproses menggunakan modulus q. Dalam prosesnya, perhitungan polinomial akan dikalikan, kemudian hasilnya direduksi sehingga diperoleh hasil akhir dari proses perhiungannya. Proses perkalian inilah yang menjadi inti dari perhitungan NTRU, karena algoritma ini sangat bergantung pada bagaimana perkalian polinomial dilakukan untuk menentukan kecepatan komputasi. Setiap parameter dalam algoritma NTRU memiliki peran matematis yang memengaruhi struktur perhitungan dan kompleksitas komputasi. Parameter n menunjukkan derajat polinomial yang digunakan dalam sistem, yang menentukan jumlah koefesien dalam setiap polinomial. Parameter q berfungsi sebagai modulus yang digunakan untuk membatasi nilai setiap koefesien dalam polinomial agar tetap berada pada ruang $[0, q - 1]$. Sementara itu, parameter p berperan sebagai modulus reduksi tambahan yang digunakan dalam proses dekripsi untuk mengembalikan hasil proses perhitungan menjadi representasi pesan. Pada akhirnya, parameter n, q dan p dalam algoritma NTRU menentukan struktur ruang polinomial tempat seluruh operasi dilakukan.



Gambar 3.1 Metode Perkalian

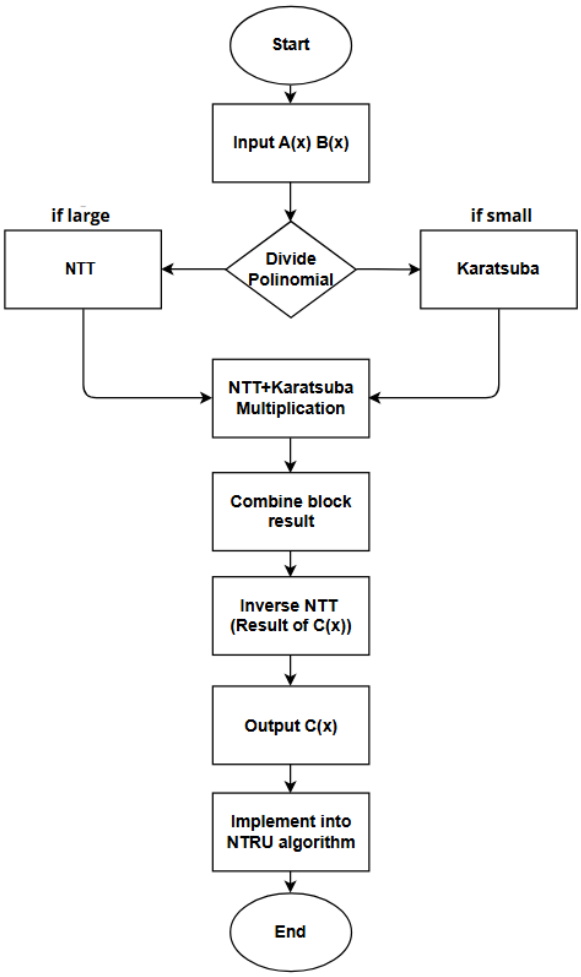
Gambar 3.1(a) adalah metode yang digunakan oleh (Zhou et al., 2018) sebagai bentuk optimalisasi proses komputasi dengan pendekatan *Preprocess-then-NTT (PtNTT)*. Metode ini dirancang untuk mengurangi kebutuhan modulus besar pada perhitungan NTT konvensional melalui tahap pra-pemrosesan polinomial sebelum dilakukan transformasi utama. Pada tahap tersebut, polinomial diuraikan menjadi dua bagian, yaitu komponen genap (even) dan komponen ganjil (odd), sehingga memungkinkan penggunaan modulus yang lebih kecil, seperti $q \equiv 1 \pmod{n}$ atau $q \equiv 1 \pmod{n/2}$. Secara matematis, pendekatan ini mengubah pola perhitungan pada transformasi *Fourier* diskret di domain bilangan bulat dan

menghasilkan bentuk representasi polinomial yang lebih terstruktur. Dampaknya, ukuran hasil komputasi seperti kunci publik dan *ciphertext* menjadi lebih kecil, meskipun terdapat tambahan proses pada tahap pra-pemrosesan dibandingkan dengan penerapan NTT standar.

Gambar 3.1(b) menunjukan pendekatan Karatsuba yang dikombinasikan dengan NTT (Zhu et al., 2019). Polinomial A(x) dan B(x) dibagi menjadi dua bagian, yaitu $A(x) = A_0 + x^{n/2}A1$ dan $B(x) = B_0 + x^{n/2}B1$. Selanjutnya diterapkan NTT pada masing-masing bagian, lalu dihitung tiga hasil utama: $P_1 = A_0B_0$, $P_2 = A_1B_1$ dan $P3 = (A_0 + A_1)(B_0 + B_1)$. Kombinasi hasil tersebut dirangkai menjadi polinomial akhir C(x) dengan formula Karatsuba. Setelah inverse NTT, polinomial C(x) dikeluarkan sebagai hasil akhir.

Sementara itu, Gambar 3.1(c) menggunakan metode yang mengombinasikan pendekatan NTT dengan algoritma Karatsuba (Kundu et al., 2022). Dalam skema ini, NTT digunakan untuk perkalian polinomial berdimensi besar, sementara Karatsuba dimanfaatkan untuk perkalian blok-blok kecil. Pendekatan hibrid ini tidak hanya memungkinkan penggunaan modulus yang lebih kecil, tetapi juga menghasilkan peningkatan proses komputasi karena kompleksitas perkalian berkurang menjadi lebih mendekati $O(n\ log\ n)$.

Adapun metode yang digunakan pada penelitian ini seperti gambar 3.2.

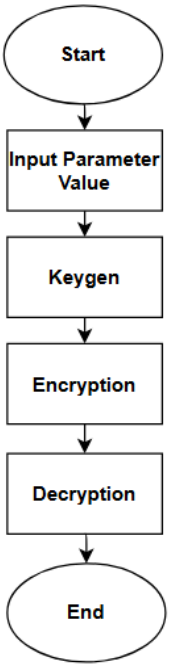


Gambar 3.2 Metode Yang Digunakan

Gambar 3.2 menunjukkan pendekatan hibrid Karatsuba–NTT yang menggabungkan dua metode perkalian polinomial untuk mempercepat proses perkalian polinomial. Proses diawali dengan memasukkan polinomial $A(x)$ dan $B(x)$, yang akan dikalikan. Kedua polinomial ini kemudian dibagi menjadi beberapa bagian (blok) untuk menyesuaikan metode perkalian yang akan digunakan. Jika ukuran blok besar, maka digunakan NTT untuk mempercepat proses konvolusi polinomial, sedangkan jika ukuran blok kecil, digunakan Karatsuba karena adanya keterbatasan besaran nilai parameter untuk melakukan

proses perhitungannya. Setelah pembagian dilakukan, proses dilanjutkan dengan perkalian pada setiap blok menggunakan kombinasi NTT dan Karatsuba, di mana hasil-hasil parsial dari tiap blok kemudian digabungkan menjadi satu bentuk polinomial utuh. Tahap berikutnya adalah melakukan Inverse NTT untuk mengubah hasil perkalian dari domain transformasi kembali ke domain koefisien, menghasilkan polinomial akhir $C(x)$. Polinomial $C(x)$ ini merupakan hasil dari perkalian $A(x)$ dan $B(x)$ menggunakan pendekatan hibrid. Hasil tersebut kemudian diintegrasikan ke dalam algoritma NTRU untuk digunakan pada proses pembangkitan kunci, enkripsi, dan dekripsi.

Proses implementasi algoritma NTRU digambarkan melalui diagram alir yang menunjukkan urutan proses algoritma bekerja yang dapat dilihat pada Gambar 3.3.

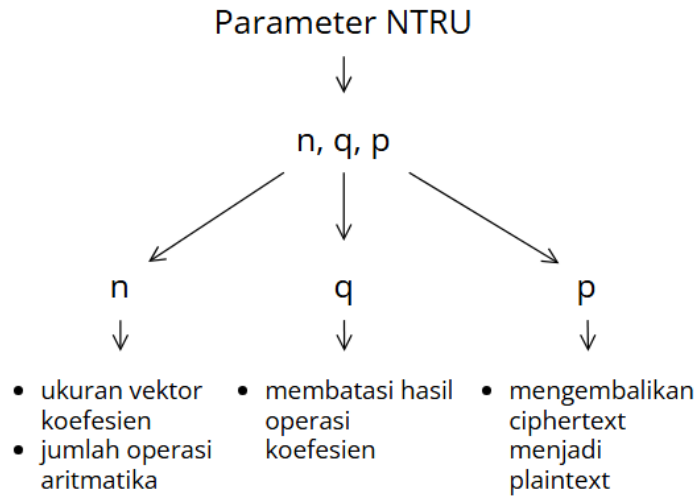


Gambar 3.3 Flowchart Algoritma NTRU

Gambar 3.3 menunjukan alur cara kerja algoritma NTRU, mulai dari tahap penentuan parameter hingga proses dekripsi yang dijelaskaan secara rinci sebagai berikut.

A. Input Parameter Value

Proses ini merupakan tahap awal dalam proses implementasi algoritma NTRU yang berfungsi untuk menentukan nilai parameter yang digunakan dalam seluruh proses. Setiap parameter memiliki perannya masing-masing seperti yang ditunjukkan pada Gambar 3.3.



Gambar 3.4 Fungsi Parameter

Parameter n menunjukkan orde atau derajat maksimum polinomial yang digunakan dalam operasi pada $Z_q[x]/(x^n - 1)$. Nilai n menentukan ukuran vektor koefisien pada setiap polinomial, semakin besar n maka semakin panjang polinomial yang diproses. Dalam konteks komputasi, nilai n memengaruhi jumlah operasi aritmetika yang dilakukan pada setiap tahap perhitungan. Apabila n=1024, maka setiap polinomial seperti $f(x)$ atau $g(x)$ memiliki 1024 koefesien, misalnya:

$$f(x) = f_0 + f_1x + f_2x^2 + \cdots + f_{1023}x^{1023}$$

Artinya, semua operasi dilakukan dengan modulo $(x^{1024} - 1)$.

Lalu, Parameter q digunakan sebagai modulus utama pada semua operasi polinomial selama proses pembangkitan kunci, enkripsi, dan dekripsi. Nilai q berfungsi untuk membatasi hasil operasi koefisien agar tetap berada dalam rentang tertentu, yaitu $0 \leq a_i \leq q$. Apabila q=83969, maka setiap koefesien hasil operasi dihitung dengan $c_i = (a_i + b_i)$ mod 83969. Sehingga semua nilai koefesien tetap berada dalam rentang $[0 - 83969]$.

Parameter p berperan saat proses dekripsi, terutama untuk mengembalikan pesan asli dari ciphertext hasil reduksi terhadap q seperti berikut.

$$m(x) = (p \cdot r(x) * h(x) + m(x)) \bmod q \bmod p$$

artinya, hasil perhitungan polinomial akan direduksi dua kali terhadap nilai q, lalu nilai p.

B. Keygen

Bagian ini merupakan tahap untuk pembangkitan kunci publik dan kunci privat yanga digunakan dalam algoritma NTRU. Pada proses ini, sistem menghasilkan polinomial utama yang disebut $f(x)$ dan $g(x)$ sebagai komponen dasar kunci. Polinomial $f(x)$ dipilih sedemikian rupa agar memiliki invers terhadap dua modulus yang berbeda, yaitu modulo p dan q, sedangkan $g(x)$ dipilih secara acak dari himpunan polinomial dengan koefisien kecil. Selanjutnya, dilakukan perhitungan untuk membentuk kunci publik (h(x)) menggunakan rumus:

$$h(x) = p \text{ x } g(x) \text{ x } f_q^{-1}(x) \text{ mod q}$$

$f_q^{-1}(x)$ adalah invers dari $f(x)$ dalam modulo q. Kunci publik $h(x)$ nantinya akan digunakan untuk proses enkripsi, sedangkan kunci privat $f(x)$ dan inversnya digunakan untuk dekripsi. Tahap ini akan menentukan nilai bit keamanan dan tingkat kompleksitas perhitungannya.

C. Enkripsi

Enkripsi merupakan tahap di mana pesan asli (plaintext) diubah menjadi bentuk terenkripsi (ciphertext) menggunakan kunci publik yang telah dihasilkan pada tahap *Keygen*. Proses ini diawali dengan mengubah pesan $m(x)$ menjadi polinomial dalam ring $\mathbb{Z}_p[x]/(x^n - 1)$. Selanjutnya, pembangkit bilangan acak $r(x)$ dipilih dari himpunan polinomial dengan koefisien kecil untuk memberikan elemen acak pada proses enkripsi, sehingga hasil ciphertext berbeda meskipun pesan yang dienkripsi sama. Setelah itu, proses enkripsi dilakukan dengan menggunakan rumus:

$$e(x) = r(x) * h(x) + m(x) \text{ (mod q)}$$

Hasil dari perhitungan ini adalah *ciphertext* $e(x)$, yaitu representasi terenkripsi dari pesan asli. *Ciphertext* ini kemudian dapat dikirimkan ke penerima tanpa khawatir isi pesan diketahui pihak lain, karena hanya pihak yang memiliki kunci privat $f(x)$ yang dapat mengembalikannya ke bentuk semula pada tahap dekripsi. Maka demikian, bagian *Encryption* berfungsi sebagai langkah utama dalam menjaga kerahasiaan data menggunakan struktur polinomial pada sistem NTRU.

D. Dekripsi

Bagian ini merupakan tahap untuk mengembalikan *cipherteks* menjadi *plaintext* menggunakan kunci privat yang telah dihasilkan pada proses keygen. Proses ini dilakukan oleh penerima pesan yang memiliki kunci privat $f(x)$. Tahap pertama dalam dekripsi adalah menghitung polinomial antara *ciphertext* $e(x)$ dan kunci privat $f(x)$ dengan operasi modulo q, menggunakan rumus:

$$a(x) = f(x) * e(x) \text{ (mod q)}$$

Hasil $a(x)$ kemudian direduksi ke dalam rentang koefisien kecil agar kembali ke domain $\mathbb{Z}_p$. Setelah itu, dilakukan proses perkalian dengan invers dari $f(x)$ terhadap modulo p, yaitu $f_p^{-1}(x)$, untuk mendapatkan pesan asli dalam bentuk polinomial bisa menggunakan rumus:

$$m(x) = f_p^{-1}(x) * a(x) \text{ (mod p)}$$

Hasil dari perhitungan ini adalah *plaintext* $m(x)$ yang sama dengan pesan awal sebelum dienkripsi. Proses dekripsi menggunakan hubungan matematis antara kunci public, kunci privat, dan parameter modulus p serta q untuk mengembalikan informasi ke bentuk semula. Bagian *Decryption* ini menjadi tahap akhir dari sistem NTRU yang memastikan hanya penerima sah dengan kunci privat yang dapat membaca pesan terenkripsi.