Adapun langkah-langkah yang digunakan pada algoritma RSA dengan menggunakan Teorema Eratothesnes adalah sebagai berikut ini:

- 1. Mendefinisikan fungsi untuk mengecek bilangan prima. Pada fungsi yang didefinisikan tersebut, untuk mengecek bilangan prima adalah dengan menggunakan Teorema Eratothesnes. Melalui Teorema Eratothesnes, akan dicek apakah terdapat  $\beta$  yang merupakan faktor suatu bilangan  $\alpha$  dimana  $\beta \leq \sqrt{\alpha}$  sehingga tidak perlu dicek hingga  $\left\lfloor \frac{\alpha}{2} \right\rfloor + 1$  yang secara bilangan lebih besar dibandingkan dengan  $\sqrt{\alpha}$ .
- 2. Mendefinisikan fungsi untuk membangkitkan bilangan prima dengan k digit yang terletak antara batas bawah dan batas atas interval bilangan prima tersebut atau dapat dinyatakan dalam interval:

$$batas\ bawah \le prima \le batas\ atas$$
 (2.1)

Pada bilangan prima dengan *k* digit, adapun batas bawah bilangan prima tersebut adalah:

$$batas\ bawah = 10^{k-1} \tag{2.2}$$

Persamaan tersebut didapatkan berdasarkan bahwa bilangan yang memiliki k digit terkecil adalah 1000 ... 000 dengan 1 digit angka 1 dan k-1 digit angka 0 atau bisa ditulis dengan  $10^{k-1}$  yang sesuai dengan persamaan (2.2). Selain itu batas atas interval dapat dinyatakan dengan:

$$batas\ atas = 10^k - 1\tag{2.3}$$

Persamaan tersebut didapatkan berdasarkan bahwa bilangan yang memiliki k digit terbesar adalah 999 ... 999 dengan k digit angka 9. Perlu diketahu bahwa bilangan bulat yang tepat berada di atas 999 ... 999 dengan selisih 1 adalah bilangan 1000 ... 000 dengan k digit angka 0 dan 1 digit angka 1 atau bisa ditulis dengan  $10^k$ . Oleh karena itu batas atas bilangan tersebut juga dapat ditulis dalam persamaan (2.3).

- 3. Menetapkan banyak digit bilangan prima yang diinginkan
- 4. Membangkitkan bilangan prima p dan q yang berbeda dengan menggunakan fungsi yang telah didefinisikan pada tahap 2
- 5. Mendefinisikasan hasil variabel baru yaitu:

$$N_1 = pq (2.4)$$

$$N_2 = (p-1)(q-1) \tag{2.5}$$

- 6. Membangkitkan public key e dengan  $gcd(e, N_2) = 1$
- 7. Membangkitkan private key d dengan  $ed \equiv 1 \mod N_2$
- 8. Menampilkan hasil public key e, private key d,  $N_1$ ,  $N_2$ , p, dan q pada program
- 9. Memasukkan plaintext yang akan dienkripsi dengan algoritma RSA
- 10. Mengubah setiap karakter plaintext menjadi decimal yang dapat diwakilkan dengan simbol *m*
- 11. Mengubah setiap plaintext dalam bentuk decimal menjadi setiap karakter dalam ciphertext *c* dengan operasi:

$$c \equiv m^e mod N_1 \tag{2.6}$$

- 12. Melalui operasi dari persamaan (2.6) didapatkan hasil enkripsi text
- 13. Untuk melakukan dekripsi dari hasil enkripsi dapat dilakukan dengan operasi:

$$m \equiv c^d mod N_1 \tag{2.7}$$

14. Melalui operasi dari persamaan (2.7) didapatkan hasil dekripsi text