A close-up of a paper with math equations

Description automatically generated

2. Konstruksi algoritma untuk mencari jarak terdekat antara 2 elemen array bilangan :

public static int MinDistance(int[] A){  
 // inisialisasi min dengan selisih elemen pertama dan kedua  
 int min = Math.*abs*(A[0] - A[1]); //Math.abs() adalah nilai absolut dari suatu bilangan  
 for(int i = 0; i < A.length; i++){  
 for(int j = i+1; j < A.length; j++){  
 if(Math.*abs*(A[i] - A[j]) < min){ //Apabila selisih elemen i dan j lebih kecil dari min  
 min = Math.*abs*(A[i] - A[j]); //Maka min diisi dengan selisih elemen i dan j  
 }  
 }  
 }  
 return min;  
}

Langkah-langkah

1. Inisialisasi variable int min dengan nilai awal jarak elemen indeks 0 dan 1
2. Lakukan perulangan untuk perbandingan jarak dengan indeks i = 0 sampai elemen terakhir, dibandingkan dengan min, apabila lebih kecil dari min, maka replace min dengan jarak
3. Lakukan perulangan hingga selesai

3. Buatlah coding dengan menggunakan algoritma binary search untuk mendapat nilai pembulatan dan akar bilangan bulat antara 0 sampai 0. Gunakan method int Findsqrt(int n) untuk menampilkan output

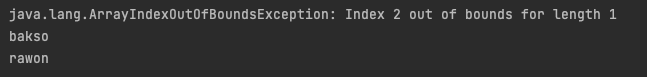
public static int findsqrt(int n){  
 if (n == 0 || n == 1) {  
 return n;  
 }  
 int start = 0, end = n, ans = 0;  
 while (start <= end) {  
 int mid = (start + end) / 2;  
 if (mid \* mid == n) {  
 return mid;  
 }  
 if (mid \* mid < n) {  
 start = mid + 1;  
 ans = mid;  
 } else {  
 end = mid - 1;  
 }  
 }  
 return ans;  
}

5. Apa output dari program ?

1. Jika statement1 benar dan statement2 benar

String a;  
String[] nama = new String[1];  
try{  
 //statement1  
 nama[2] = "Lindia";  
 //statement2  
}catch (ArithmeticException ex1) {  
 System.*out*.println(ex1);  
}catch (Exception ex2) {  
 System.*out*.println(ex2);  
} finally {  
 System.*out*.println("bakso");  
}  
System.*out*.println("rawon");

Output :



1. Jika statement1 adalah a=3 dan statement2 benar

String a;  
String[] nama = new String[1];  
try{  
 a = 3;  
 nama[2] = "Lindia";  
 //statement2  
}catch (ArithmeticException ex1) {  
 System.*out*.println(ex1);  
}catch (Exception ex2) {  
 System.*out*.println(ex2);  
} finally {  
 System.*out*.println("bakso");  
}  
System.*out*.println("rawon");

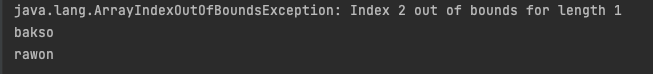
Output :



1. Jika statement1 benar, dan statement2 adalah sout(1/0);

String a;  
String[] nama = new String[1];  
try{  
 //statement1  
 nama[2] = "Lindia";  
 System.*out*.println(1/0);  
}catch (ArithmeticException ex1) {  
 System.*out*.println(ex1);  
}catch (Exception ex2) {  
 System.*out*.println(ex2);  
} finally {  
 System.*out*.println("bakso");  
}  
System.*out*.println("rawon");

Output :



1. Jika statement1 diganti sout(1/0) dan statement2 benar

String a;  
String[] nama = new String[1];  
try{  
 System.*out*.println(1/0);  
 nama[2] = "Lindia";  
 //statement2  
}catch (ArithmeticException ex1) {  
 System.*out*.println(ex1);  
}catch (Exception ex2) {  
 System.*out*.println(ex2);  
} finally {  
 System.*out*.println("bakso");  
}  
System.*out*.println("rawon");

Output :

A black background with white text

Description automatically generated