Tugas 3 PBO Makassar, 05 Maret 2024

TUGAS PEMROGRAMAN

BEROIENTASI OBJEK



Nama : Muh Rusman

Stambuk : 13020220068

Dosen : Mardiyyah Hasnawi, S.Kom.,M.T.

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

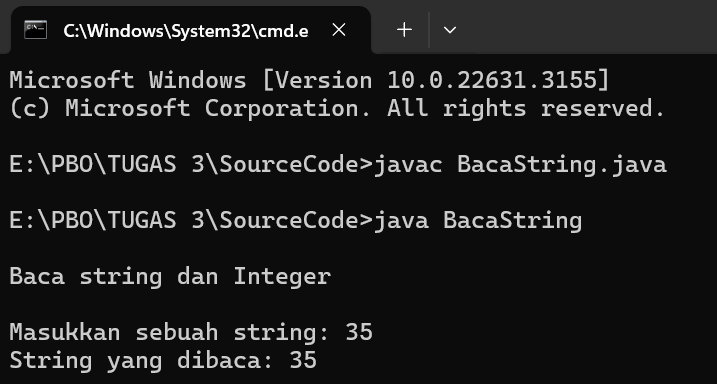
UNIVERSITAS MUSLIM INDONESIA

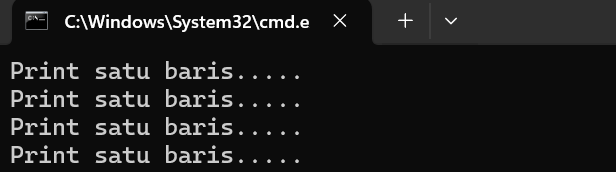
MAKASSAR

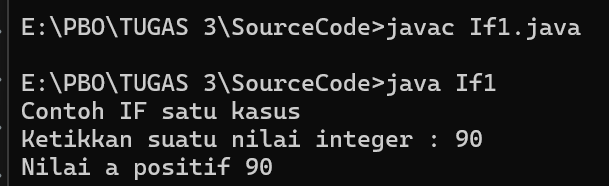
2024

**1.BacaString**

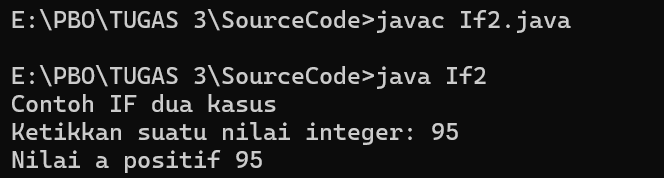
1. import java.io.BufferedReader; - Kita mengimpor kelas BufferedReader dari paket java.io. BufferedReader adalah sebuah kelas yang digunakan untuk membaca karakter dari sebuah sumber input (dalam kasus ini, System.in yang merupakan input standar). BufferedReader menggunakan buffer untuk meningkatkan kinerja saat membaca karakter.
2. import java.io.IOException; - Kita mengimpor kelas IOException dari paket java.io. IOException adalah sebuah kelas yang merepresentasikan kesalahan saat melakukan operasi input/output.
3. public class BacaString { - Kita membuat kelas bernama BacaString.
4. public static void main(String[] args) throws IOException { - Metode main adalah metode yang akan dijalankan saat program dijalankan. throws IOException menunjukkan bahwa metode ini mungkin melemparkan sebuah IOException.
5. String str; - Kita membuat sebuah variabel bernama str yang akan menyimpan string yang dibaca.
6. BufferedReader dataIn = new BufferedReader(new InputStreamReader(System.in)); - Kita membuat sebuah objek BufferedReader bernama dataIn. BufferedReader dibangun di atas InputStreamReader yang dibangun di atas System.in. System.in adalah sebuah input stream yang menyediakan akses ke input standar.
7. System.out.println("\nBaca string dan Integer \n"); - Kita mencetak pesan yang menginstruksikan pengguna untuk memasukkan sebuah string.
8. System.out.print("Masukkan sebuah string: "); - Kita mencetak pesan yang menginstruksikan pengguna untuk memasukkan sebuah string.
9. str = dataIn.readLine(); - Kita menggunakan metode readLine() dari dataIn untuk membaca string yang diinputkan oleh pengguna. Metode ini akan membaca karakter hingga menemukan karakter newline atau akhir file. Setelah itu, metode ini akan mengembalikan string yang dibaca.
10. System.out.println("String yang dibaca: " + str); - Kita mencetak pesan yang menunjukkan string yang telah dibaca.
11. } - Penutup kelas BacaString.

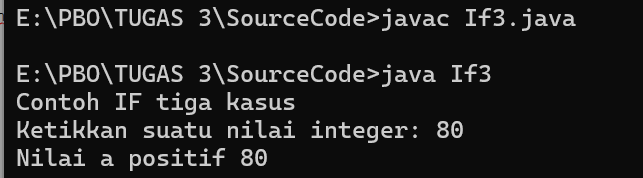


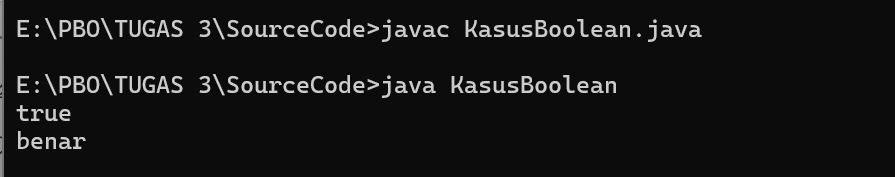
1. **ForEver**
2. public class ForEver {: Ini mendeklarasikan sebuah kelas publik dengan nama ForEver.
3. public static void main(String[] args) {: Ini adalah metode utama (main method), yang berfungsi sebagai titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan mengambil sebuah array dari string sebagai argumennya.
4. System.out.println("Program akan looping, akhiri dengan ^c");: Baris ini mencetak pesan "Program akan looping, akhiri dengan ^c" ke konsol, yang berarti "Program akan terus melakukan loop, berhenti dengan ^c".
5. while (true) {: Ini adalah awal dari sebuah loop tak terhingga, karena kondisi true akan selalu bernilai true.
6. System.out.println("Print satu baris.....");: Baris ini mencetak pesan "Print satu baris....." ke konsol.
7. }: Kurung penutup untuk loop tak terhingga.
8. }: Kurung penutup untuk metode utama.
9. }: Kurung penutup untuk kelas.
10. **If1**
11. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
12. public class If1 {: Ini mendeklarasikan sebuah kelas publik dengan nama If1.
13. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas If1, yang berfungsi sebagai titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
14. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
15. int a;: Ini mendeklarasikan sebuah variabel bertipe integer dengan nama a.
16. System.out.println("Contoh IF satu kasus");: Baris ini mencetak pesan "Contoh IF satu kasus" ke konsol.
17. System.out.print("Ketikkan suatu nilai integer : ");: Baris ini mencetak pesan "Ketikkan suatu nilai integer : " tanpa karakter newline ke konsol.
18. a = masukan.nextInt();: Baris ini menggunakan objek masukan untuk membaca sebuah nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel a.
19. if (a >= 0) {: Baris ini memulai blok if yang mengevaluasi apakah nilai yang dimasukkan pengguna ke variabel a adalah non-negatif (lebih besar dari atau sama dengan nol).
20. System.out.println("Nilai a positif " + a);: Jika kondisi di atas benar (nilai a non-negatif), baris ini mencetak pesan "Nilai a positif" diikuti dengan nilai dari variabel a ke konsol.
21. }: Kurung penutup untuk blok if.
22. }: Kurung penutup untuk metode utama.



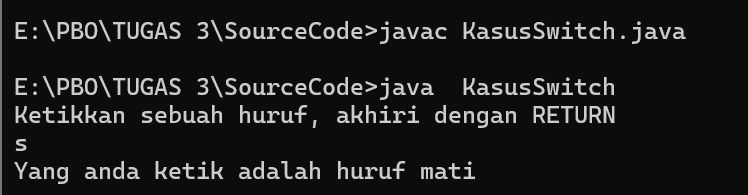
1. **If2**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class If2 {: Ini mendeklarasikan sebuah kelas publik dengan nama If2.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas If2, yang berfungsi sebagai titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. int a;: Ini mendeklarasikan sebuah variabel bertipe integer dengan nama a.
6. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
7. System.out.println("Contoh IF dua kasus");: Baris ini mencetak pesan "Contoh IF dua kasus" ke konsol.
8. System.out.print("Ketikkan suatu nilai integer: ");: Baris ini mencetak pesan "Ketikkan suatu nilai integer: " tanpa karakter newline ke konsol.
9. a = masukan.nextInt();: Baris ini menggunakan objek masukan untuk membaca sebuah nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel a.
10. if (a >= 0) {: Baris ini memulai blok if yang mengevaluasi apakah nilai yang dimasukkan pengguna ke variabel a adalah non-negatif (lebih besar dari atau sama dengan nol).
11. System.out.println("Nilai a positif " + a);: Jika kondisi di atas benar (nilai a non-negatif), baris ini mencetak pesan "Nilai a positif" diikuti dengan nilai dari variabel a ke konsol.
12. } else {: Jika kondisi pada blok if di atas tidak terpenuhi, maka eksekusi akan masuk ke blok else.
13. System.out.println("Nilai a negatif " + a);: Baris ini mencetak pesan "Nilai a negatif" diikuti dengan nilai dari variabel a ke konsol.
14. }: Kurung penutup untuk blok if-else.
15. }: Kurung penutup untuk metode utama.



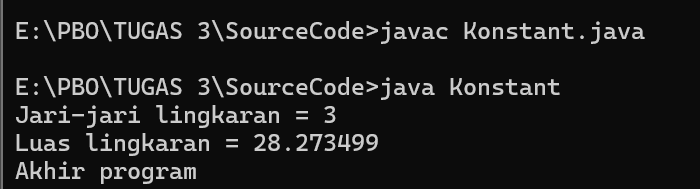
1. **If3**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class If3 {: Ini mendeklarasikan sebuah kelas publik dengan nama If3.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas If3, yang berfungsi sebagai titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. int a;: Ini mendeklarasikan sebuah variabel bertipe integer dengan nama a.
6. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
7. System.out.println("Contoh IF tiga kasus");: Baris ini mencetak pesan "Contoh IF tiga kasus" ke konsol.
8. System.out.print("Ketikkan suatu nilai integer: ");: Baris ini mencetak pesan "Ketikkan suatu nilai integer: " tanpa karakter newline ke konsol.
9. a = masukan.nextInt();: Baris ini menggunakan objek masukan untuk membaca sebuah nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel a.
10. if (a > 0) {: Baris ini memulai blok if yang mengevaluasi apakah nilai yang dimasukkan pengguna ke variabel a adalah positif.
11. System.out.println("Nilai a positif " + a);: Jika kondisi di atas benar (nilai a positif), baris ini mencetak pesan "Nilai a positif" diikuti dengan nilai dari variabel a ke konsol.
12. } else if (a == 0) {: Jika kondisi pada blok if di atas tidak terpenuhi, maka eksekusi akan masuk ke blok else if ini, yang mengecek apakah nilai a sama dengan nol.
13. System.out.println("Nilai Nol " + a);: Jika kondisi pada blok else if di atas benar (nilai a sama dengan nol), baris ini mencetak pesan "Nilai Nol" diikuti dengan nilai dari variabel a ke konsol.
14. } else {: Jika kondisi pada blok else if di atas tidak terpenuhi, maka eksekusi akan masuk ke blok else, yang menangkap semua kasus ketika nilai a kurang dari nol.
15. System.out.println("Nilai a negatif " + a);: Baris ini mencetak pesan "Nilai a negatif" diikuti dengan nilai dari variabel a ke konsol.
16. }: Kurung penutup untuk blok if-else if-else.
17. }: Kurung penutup untuk metode utama.
18. **KasusBoolean**
19. public class KasusBoolean {: Ini mendeklarasikan sebuah kelas publik dengan nama KasusBoolean.
20. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas KasusBoolean, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
21. boolean bool;: Mendeklarasikan sebuah variabel dengan tipe data boolean dengan nama bool.
22. bool = true;: Menginisialisasi variabel bool dengan nilai true.
23. if (bool) {: Memulai blok if yang mengevaluasi nilai dari variabel bool. Jika nilai bool adalah true, maka eksekusi akan masuk ke dalam blok ini.
24. System.out.print("true\n");: Jika kondisi pada blok if di atas benar (nilai bool adalah true), maka program akan mencetak pesan "true" diikuti dengan karakter newline ke konsol.
25. } else {: Jika kondisi pada blok if di atas tidak terpenuhi, maka eksekusi akan masuk ke dalam blok else.
26. System.out.print("false\n");: Jika kondisi pada blok if di atas tidak terpenuhi (nilai bool bukan true), maka program akan mencetak pesan "false" diikuti dengan karakter newline ke konsol.
27. if (!bool) {: Memulai blok if yang mengevaluasi kebalikan dari nilai dari variabel bool dengan operator ! (negasi). Jika nilai bool adalah false, maka eksekusi akan masuk ke dalam blok ini.
28. System.out.print("salah\n");: Jika kondisi pada blok if di atas benar (nilai bool adalah false), maka program akan mencetak pesan "salah" diikuti dengan karakter newline ke konsol.
29. } else {: Jika kondisi pada blok if di atas tidak terpenuhi, maka eksekusi akan masuk ke dalam blok else.
30. System.out.print("benar\n");: Jika kondisi pada blok if di atas tidak terpenuhi (nilai bool bukan false), maka program akan mencetak pesan "benar" diikuti dengan karakter newline ke konsol.
31. }: Kurung penutup untuk blok if-else.
32. }: Kurung penutup untuk metode utama.



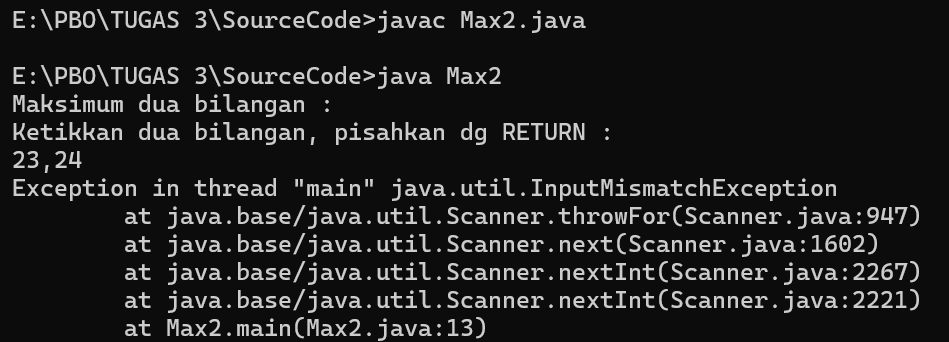
1. **KasusSwitch**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class KasusSwitch {: Ini mendeklarasikan sebuah kelas publik dengan nama KasusSwitch.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas KasusSwitch, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. char cc;: Mendeklarasikan sebuah variabel dengan tipe data karakter (char) dengan nama cc.
6. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
7. System.out.print("Ketikkan sebuah huruf, akhiri dengan RETURN \n");: Baris ini mencetak pesan "Ketikkan sebuah huruf, akhiri dengan RETURN" ke konsol.
8. cc = masukan.next().charAt(0);: Baris ini menggunakan objek masukan untuk membaca karakter yang dimasukkan oleh pengguna dan menyimpannya dalam variabel cc.
9. switch (cc) {: Memulai struktur kontrol switch dengan ekspresi cc, yang akan mengevaluasi nilai dari variabel cc.
10. case 'a': {: Jika nilai dari variabel cc adalah 'a', maka eksekusi akan masuk ke dalam blok case ini.
11. System.out.print("Yang anda ketik adalah a\n");: Baris ini mencetak pesan "Yang anda ketik adalah a" ke konsol.
12. break;: Kata kunci break digunakan untuk keluar dari struktur switch setelah blok case yang sesuai sudah dieksekusi.
13. case 'u': {: Ini adalah blok case kedua, yang mengecek apakah nilai dari variabel cc adalah 'u'.
14. System.out.print("Yang anda ketik adalah u\n");: Jika nilai dari variabel cc adalah 'u', maka program akan mencetak pesan "Yang anda ketik adalah u" ke konsol.
15. default:: Ini adalah bagian default dari struktur switch, yang akan dieksekusi jika tidak ada blok case yang cocok dengan nilai dari variabel cc.
16. System.out.print("Yang anda ketik adalah huruf mati\n");: Jika nilai dari variabel cc tidak cocok dengan nilai dalam blok case sebelumnya, maka program akan mencetak pesan "Yang anda ketik adalah huruf mati" ke konsol.
17. }: Kurung penutup untuk struktur switch.
18. }: Kurung penutup untuk metode utama.

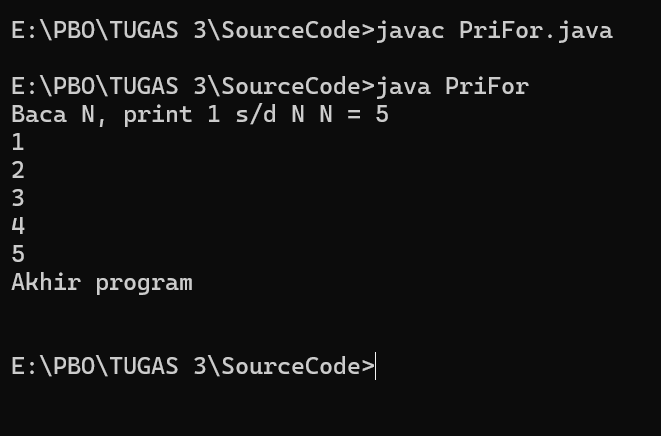
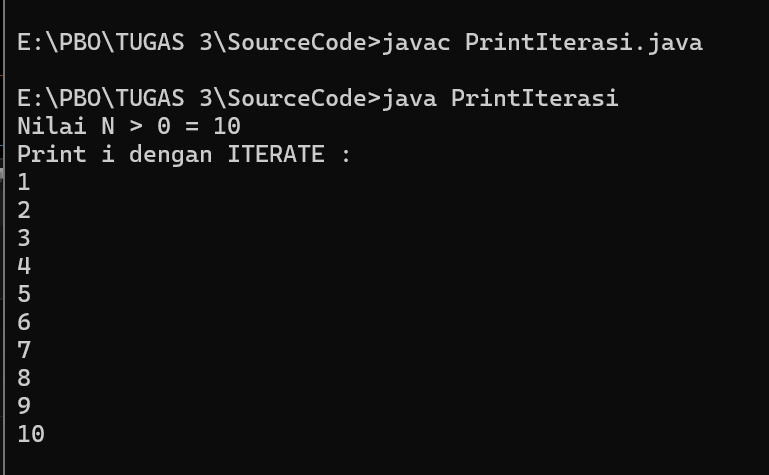


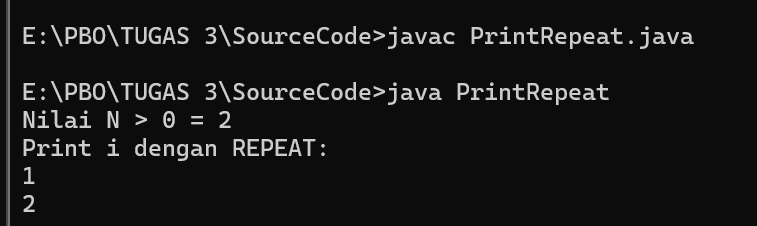
1. **Konstant**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class Konstant {: Ini mendeklarasikan sebuah kelas publik dengan nama Konstant.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas Konstant, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. final float PHI = 3.1415f;: Mendeklarasikan sebuah variabel konstanta PHI yang bertipe data float dan diinisialisasi dengan nilai 3.1415. Kata kunci final digunakan untuk membuat variabel konstanta yang nilainya tidak dapat diubah setelah diinisialisasi.
6. float r;: Mendeklarasikan sebuah variabel bertipe data float dengan nama r, yang akan digunakan untuk menyimpan nilai jari-jari lingkaran yang dimasukkan oleh pengguna.
7. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
8. System.out.print("Jari-jari lingkaran = ");: Baris ini mencetak pesan "Jari-jari lingkaran = " tanpa karakter newline ke konsol, untuk meminta pengguna memasukkan nilai jari-jari lingkaran.
9. r = masukan.nextFloat();: Baris ini menggunakan objek masukan untuk membaca sebuah nilai floating-point yang dimasukkan oleh pengguna dan menyimpannya dalam variabel r.
10. System.out.println("Luas lingkaran = " + (PHI \* r \* r));: Baris ini mencetak pesan "Luas lingkaran = " diikuti dengan hasil perhitungan luas lingkaran (π \* r \* r) ke konsol. Variabel PHI digunakan untuk merepresentasikan nilai konstanta π, dan r adalah nilai jari-jari yang dimasukkan pengguna.
11. System.out.println("Akhir program");: Baris ini mencetak pesan "Akhir program" ke konsol, menandakan akhir dari program.
12. }: Kurung penutup untuk metode utama.
13. }: Kurung penutup untuk kelas Konstant.



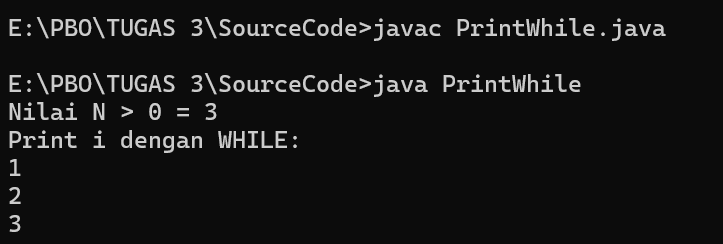
1. **Max2**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class Max2 {: Ini mendeklarasikan sebuah kelas publik dengan nama Max2.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas Max2, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. int a, b;: Mendeklarasikan dua variabel bertipe data integer dengan nama a dan b, yang akan digunakan untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.
6. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
7. System.out.print("Maksimum dua bilangan : \n");: Baris ini mencetak pesan "Maksimum dua bilangan : " tanpa karakter newline ke konsol.
8. System.out.print("Ketikkan dua bilangan, pisahkan dg RETURN :\n");: Baris ini mencetak pesan "Ketikkan dua bilangan, pisahkan dg RETURN :" tanpa karakter newline ke konsol, untuk meminta pengguna memasukkan dua bilangan bulat.
9. a = masukan.nextInt();: Baris ini menggunakan objek masukan untuk membaca bilangan bulat pertama yang dimasukkan oleh pengguna dan menyimpannya dalam variabel a.
10. b = masukan.nextInt();: Baris ini menggunakan objek masukan untuk membaca bilangan bulat kedua yang dimasukkan oleh pengguna dan menyimpannya dalam variabel b.
11. System.out.println("Ke dua bilangan : a = " + a + " b = " + b);: Baris ini mencetak pesan yang menampilkan kedua bilangan yang dimasukkan oleh pengguna ke konsol.
12. if (a >= b) {: Memulai blok if yang mengevaluasi apakah nilai a lebih besar dari atau sama dengan nilai b. Jika kondisi ini benar, maka eksekusi akan masuk ke dalam blok if.
13. System.out.println("Nilai a yang maksimum " + a);: Jika kondisi pada blok if di atas benar (nilai a lebih besar dari atau sama dengan nilai b), maka program akan mencetak pesan yang menyatakan bahwa nilai a adalah nilai maksimum ke konsol.
14. } else /\* a > b \*/ {: Jika kondisi pada blok if di atas tidak terpenuhi (nilai a lebih kecil dari nilai b), maka eksekusi akan masuk ke dalam blok else.
15. System.out.println("Nilai b yang maksimum: " + b);: Baris ini mencetak pesan yang menyatakan bahwa nilai b adalah nilai maksimum ke konsol.
16. }: Kurung penutup untuk blok if-else.
17. }: Kurung penutup untuk metode utama.

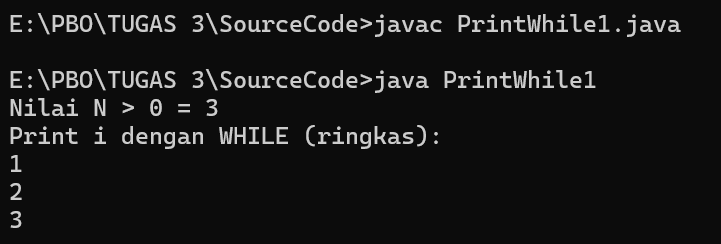
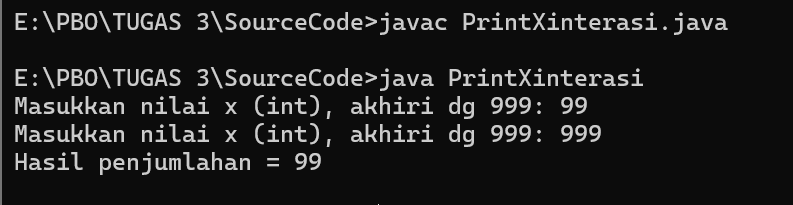


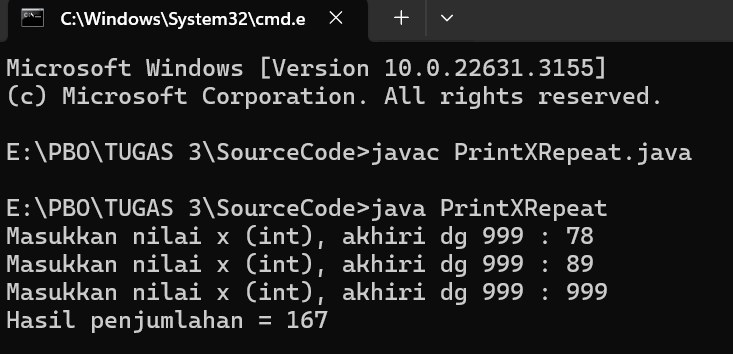
1. **PriFor**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class PriFor {: Ini mendeklarasikan sebuah kelas publik dengan nama PriFor.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas PriFor, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. int i, N;: Mendeklarasikan dua variabel bertipe data integer dengan nama i dan N. Variabel i akan digunakan sebagai variabel loop dalam perulangan for, sementara N akan menyimpan nilai yang dimasukkan oleh pengguna.
6. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
7. System.out.print("Baca N, print 1 s/d N ");: Mencetak pesan "Baca N, print 1 s/d N " tanpa karakter newline ke konsol.
8. System.out.print("N = ");: Mencetak pesan "N = " tanpa karakter newline ke konsol, untuk menandakan pengguna diminta untuk memasukkan nilai N.
9. N = masukan.nextInt();: Menggunakan objek masukan untuk membaca sebuah bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.
10. for (i = 1; i <= N; i++) {: Memulai perulangan for dengan variabel i dimulai dari 1, terus bertambah satu hingga mencapai atau melebihi nilai N.
11. System.out.println(i);: Mencetak nilai dari variabel i ke konsol dalam setiap iterasi, sehingga mencetak semua bilangan bulat dari 1 hingga N.
12. System.out.println("Akhir program \n");: Mencetak pesan "Akhir program" diikuti dengan karakter newline ke konsol, menandakan akhir dari program.
13. }: Kurung penutup untuk perulangan for.
14. }: Kurung penutup untuk metode utama.
15. **PrintIterasi**
16. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
17. public class PrintIterasi {: Ini mendeklarasikan sebuah kelas publik dengan nama PrintIterasi.
18. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas PrintIterasi, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
19. int N;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama N, yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
20. int i;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama i, yang akan digunakan sebagai variabel loop.
21. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
22. System.out.print("Nilai N > 0 = ");: Mencetak pesan "Nilai N > 0 = " tanpa karakter newline ke konsol, untuk meminta pengguna memasukkan nilai N.
23. N = masukan.nextInt();: Menggunakan objek masukan untuk membaca sebuah bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.
24. i = 1;: Menginisialisasi variabel i dengan nilai 1.
25. System.out.print("Print i dengan ITERATE : \n");: Mencetak pesan "Print i dengan ITERATE : " tanpa karakter newline ke konsol.
26. for (;;) {: Memulai perulangan for tanpa menyertakan kondisi awal, kondisi terminasi, atau pengubah langkah. Ini berarti perulangan ini akan berlangsung tanpa henti sampai ada instruksi break yang menghentikannya.
27. System.out.println(i);: Mencetak nilai dari variabel i ke konsol dalam setiap iterasi.
28. if (i == N): Memeriksa apakah nilai i sudah sama dengan nilai N.
29. break;: Jika nilai i sudah sama dengan nilai N, maka instruksi break akan menghentikan perulangan.
30. else { i++; }: Jika nilai i masih belum sama dengan nilai N, maka variabel i akan ditambah satu untuk melanjutkan iterasi berikutnya.
31. }: Kurung penutup untuk perulangan for.
32. }: Kurung penutup untuk metode utama.
33. **PrintRepeat**
34. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
35. public class PrintRepeat {: Ini mendeklarasikan sebuah kelas publik dengan nama PrintRepeat.
36. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas PrintRepeat, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
37. int N;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama N, yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
38. int i;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama i, yang akan digunakan sebagai variabel loop.
39. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
40. System.out.print("Nilai N > 0 = ");: Mencetak pesan "Nilai N > 0 = " tanpa karakter newline ke konsol, untuk meminta pengguna memasukkan nilai N.
41. N = masukan.nextInt();: Menggunakan objek masukan untuk membaca sebuah bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.
42. i = 1;: Menginisialisasi variabel i dengan nilai 1.
43. System.out.print("Print i dengan REPEAT: \n");: Mencetak pesan "Print i dengan REPEAT: " tanpa karakter newline ke konsol.
44. do {: Memulai blok perulangan do-while, yang akan menjalankan blok pernyataan terlebih dahulu sebelum memeriksa kondisi perulangan.
45. System.out.print(i + "\n");: Mencetak nilai dari variabel i ke konsol dalam setiap iterasi, diikuti dengan karakter newline.
46. i++;: Menaikkan nilai variabel i setiap kali iterasi dilakukan, sehingga mencetak semua bilangan bulat dari 1 hingga N.
47. } while (i <= N);: Memeriksa apakah nilai variabel i masih kurang dari atau sama dengan nilai N. Jika kondisi ini terpenuhi, maka perulangan akan terus dilakukan. Jika tidak, perulangan akan dihentikan.
48. }: Kurung penutup untuk blok do-while.
49. }: Kurung penutup untuk metode utama.

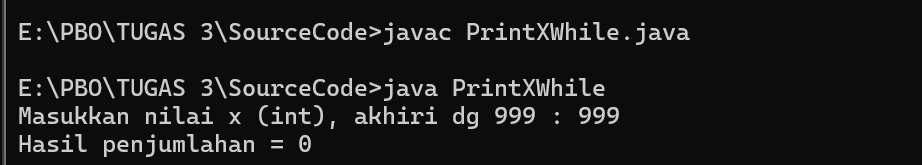


1. **PrintWhile**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class PrintWhile {: Ini mendeklarasikan sebuah kelas publik dengan nama PrintWhile.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas PrintWhile, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. int N;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama N, yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
6. int i;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama i, yang akan digunakan sebagai variabel loop.
7. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
8. System.out.print("Nilai N > 0 = ");: Mencetak pesan "Nilai N > 0 = " tanpa karakter newline ke konsol, untuk meminta pengguna memasukkan nilai N.
9. N = masukan.nextInt();: Menggunakan objek masukan untuk membaca sebuah bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.
10. i = 1;: Menginisialisasi variabel i dengan nilai 1.
11. System.out.print("Print i dengan WHILE: \n");: Mencetak pesan "Print i dengan WHILE: " tanpa karakter newline ke konsol.
12. while (i <= N) {: Memulai blok perulangan while, yang akan terus berlangsung selama kondisi i <= N terpenuhi.
13. System.out.println(i);: Mencetak nilai dari variabel i ke konsol dalam setiap iterasi.
14. i++;: Menaikkan nilai variabel i setiap kali iterasi dilakukan, sehingga mencetak semua bilangan bulat dari 1 hingga N.
15. };: Tanda titik koma diakhiri kurung kurawal menandakan akhir dari blok while.
16. }: Kurung penutup untuk metode utama.

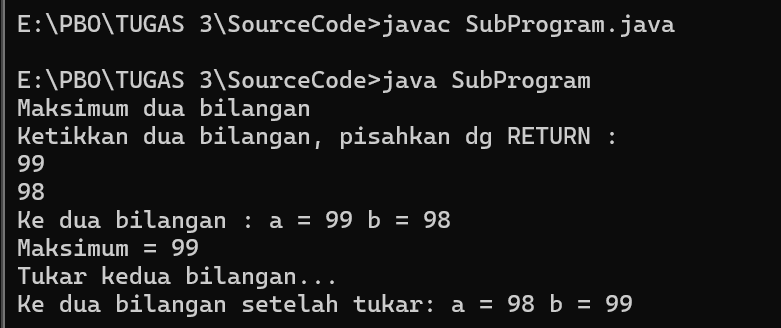


1. **PrrintWhile1**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class PrintWhile1 {: Ini mendeklarasikan sebuah kelas publik dengan nama PrintWhile1.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas PrintWhile1, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. int N;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama N, yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
6. int i = 1;: Mendeklarasikan dan menginisialisasi sebuah variabel bertipe data integer dengan nama i dengan nilai awal 1.
7. Scanner masukan = new Scanner(System.in);: Baris ini membuat objek Scanner bernama masukan, yang akan digunakan untuk membaca masukan dari pengguna melalui System.in.
8. System.out.print("Nilai N > 0 = ");: Mencetak pesan "Nilai N > 0 = " tanpa karakter newline ke konsol, untuk meminta pengguna memasukkan nilai N.
9. N = masukan.nextInt();: Menggunakan objek masukan untuk membaca sebuah bilangan bulat yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.
10. System.out.print("Print i dengan WHILE (ringkas): \n");: Mencetak pesan "Print i dengan WHILE (ringkas): " tanpa karakter newline ke konsol.
11. while (i <= N) {: Memulai blok perulangan while, yang akan terus berlangsung selama kondisi i <= N terpenuhi.
12. System.out.println(i++);: Mencetak nilai dari variabel i ke konsol dalam setiap iterasi, kemudian menaikkan nilai variabel i menggunakan operator post-increment (i++). Ini akan menyebabkan nilai i terlebih dahulu dicetak, kemudian diinkrementasi.
13. } /\* (i > N) \*/: Kurung penutup untuk blok while. Komentar ini menunjukkan akhir dari blok while, meskipun tidak diperlukan karena sudah ada tanda kurung kurawal.
14. **PrintXinterasi**
15. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
16. public class PrintXinterasi {: Mendeklarasikan kelas publik dengan nama PrintXinterasi.
17. public static void main(String[] args) {: Mendefinisikan metode utama main() untuk kelas PrintXinterasi. Ini adalah titik masuk program, di mana eksekusi program dimulai.
18. int Sum = 0;: Mendeklarasikan variabel Sum yang bertipe integer dan diinisialisasi dengan nilai 0. Variabel ini akan digunakan untuk menyimpan hasil penjumlahan dari nilai-nilai x yang dimasukkan oleh pengguna.
19. int x;: Mendeklarasikan variabel x yang bertipe integer. Variabel ini akan digunakan untuk menyimpan nilai x yang dimasukkan oleh pengguna.
20. Scanner masukan = new Scanner(System.in);: Membuat objek Scanner dengan nama masukan untuk membaca masukan dari pengguna melalui System.in.
21. System.out.print("Masukkan nilai x (int), akhiri dg 999: ");: Mencetak pesan permintaan untuk pengguna untuk memasukkan nilai x. Pengguna diminta untuk mengakhiri masukan dengan memasukkan nilai 999.
22. x = masukan.nextInt();: Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel x.
23. if (x == 999) {: Memeriksa apakah pengguna langsung memasukkan nilai 999 sebagai input.
24. System.out.print("Kasus kosong\n");: Jika pengguna memasukkan nilai 999 sebagai input, maka program akan mencetak pesan "Kasus kosong" ke konsol, karena tidak ada nilai yang dimasukkan selain nilai 999.
25. else {: Jika pengguna tidak langsung memasukkan nilai 999 sebagai input, maka program akan menjalankan pernyataan-pernyataan berikutnya.
26. Sum = x;: Menyimpan nilai x dalam variabel Sum.
27. for (;;) {: Memulai perulangan for tanpa kondisi terminasi, yang berarti perulangan akan berlangsung tanpa batas.
28. System.out.print("Masukkan nilai x (int), akhiri dg 999: ");: Mencetak pesan permintaan untuk pengguna memasukkan nilai x kembali.
29. x = masukan.nextInt();: Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel x.
30. if (x == 999): Memeriksa apakah pengguna memasukkan nilai 999.
31. break;: Jika pengguna memasukkan nilai 999, perulangan akan dihentikan.
32. else { Sum = Sum + x; }: Jika pengguna memasukkan nilai selain 999, nilai x akan ditambahkan ke dalam variabel Sum.
33. }: Kurung penutup untuk blok perulangan for.
34. System.out.println("Hasil penjumlahan = " + Sum);: Setelah perulangan selesai, program akan mencetak hasil penjumlahan dari semua nilai x yang dimasukkan pengguna ke konsol.
35. }: Kurung penutup untuk blok else.
36. }: Kurung penutup untuk metode utama.
37. **PrintXRepeat**
38. import java.util.Scanner;: Mendeklarasikan penggunaan pustaka Scanner yang terletak dalam paket java.util. Ini diperlukan untuk memungkinkan program membaca input dari pengguna.
39. public class PrintXRepeat { ... }: Mendefinisikan kelas Java utama dengan nama PrintXRepeat. Ini adalah kelas yang akan berisi metode main() dan metode lainnya jika diperlukan.
40. public static void main(String[] args) { ... }: Metode utama program, dieksekusi saat program dijalankan. Semua instruksi di dalamnya akan dijalankan secara berurutan.
41. Komentar // TODO Auto-generated method stub: Komentar ini secara otomatis dihasilkan oleh beberapa IDE (Integrated Development Environment) untuk menandai bahwa metode belum sepenuhnya diimplementasikan.
42. int Sum; int x;: Deklarasi variabel Sum dan x dengan tipe data int.
43. Scanner masukan = new Scanner(System.in);: Membuat objek Scanner bernama masukan untuk menerima input dari pengguna melalui keyboard (System.in).
44. System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");: Mengeluarkan pesan ke layar untuk meminta pengguna memasukkan nilai. Ini juga berfungsi sebagai petunjuk bahwa pengguna dapat mengakhiri proses input dengan memasukkan angka 999.
45. x = masukan.nextInt();: Membaca input pengguna sebagai nilai integer dan menyimpannya dalam variabel x.
46. if (x == 999) { ... } else { ... }: Mengecek apakah nilai x yang dimasukkan sama dengan 999. Jika ya, maka program akan mencetak "Kasus kosong" dan berakhir. Jika tidak, proses penjumlahan akan dimulai.
47. Sum = 0;: Inisialisasi variabel Sum dengan nilai 0, yang akan digunakan untuk menyimpan jumlah dari nilai-nilai yang dimasukkan oleh pengguna.
48. do { ... } while (x != 999);: Melakukan pengulangan untuk membaca nilai dari pengguna dan menambahkannya ke dalam Sum sampai pengguna memasukkan angka 999.
49. Sum = Sum + x;: Menambahkan nilai x ke dalam Sum untuk mengakumulasikan jumlah nilai.
50. System.out.println("Hasil penjumlahan = " + Sum);: Mencetak hasil akhir penjumlahan dari semua nilai yang dimasukkan pengguna.



1. **PrintXwhile**
2. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
3. public class PrintXWhile {: Mendeklarasikan kelas publik dengan nama PrintXWhile.
4. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas PrintXWhile, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
5. int Sum;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama Sum, yang akan digunakan untuk menyimpan hasil penjumlahan dari nilai-nilai x yang dimasukkan oleh pengguna.
6. int x;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama x, yang akan digunakan untuk menyimpan nilai x yang dimasukkan oleh pengguna.
7. Scanner masukan = new Scanner(System.in);: Membuat objek Scanner dengan nama masukan untuk membaca masukan dari pengguna melalui System.in.
8. Sum = 0;: Menginisialisasi variabel Sum dengan nilai 0, yang akan digunakan untuk menyimpan hasil penjumlahan nilai-nilai x.
9. System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");: Mencetak pesan permintaan untuk pengguna untuk memasukkan nilai x. Pengguna diminta untuk mengakhiri masukan dengan memasukkan nilai 999.
10. x = masukan.nextInt();: Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel x.
11. while (x != 999) {: Memulai perulangan while, yang akan terus berlangsung selama nilai x yang dimasukkan oleh pengguna bukanlah 999.
12. Sum = Sum + x;: Menambahkan nilai x ke dalam variabel Sum.
13. System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");: Mencetak pesan permintaan untuk pengguna untuk memasukkan nilai x kembali.
14. x = masukan.nextInt();: Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel x.
15. }: Kurung penutup untuk blok while.
16. System.out.println("Hasil penjumlahan = " + Sum);: Setelah perulangan selesai, program akan mencetak hasil penjumlahan dari semua nilai x yang dimasukkan oleh pengguna ke konsol.
17. }: Kurung penutup untuk metode utama.

**18. SubProgram**

1. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
2. public class SubProgram {: Mendeklarasikan kelas publik dengan nama SubProgram.
3. public static int maxab(int a, int b) {: Mendefinisikan sebuah subprogram maxab yang mengambil dua parameter bertipe integer (a dan b) dan mengembalikan nilai integer.
4. return ((a >= b) ? a : b);: Mengembalikan nilai a jika a lebih besar dari atau sama dengan b, dan mengembalikan nilai b jika tidak. Ini menggunakan operator ternary (?:) untuk menentukan nilai maksimum antara dua bilangan.
5. public static void tukar(int a, int b) {: Mendefinisikan sebuah subprogram tukar yang mengambil dua parameter bertipe integer (a dan b) dan tidak mengembalikan nilai (void).
6. int temp;: Mendeklarasikan variabel lokal temp yang bertipe integer untuk menyimpan nilai sementara.
7. temp = a; a = b; b = temp;: Menukar nilai a dan b dengan menggunakan variabel sementara temp.
8. System.out.println("Ke dua bilangan setelah tukar: a = " + a + " b = " + b);: Mencetak nilai kedua bilangan setelah dilakukan pertukaran.
9. public static void main(String[] args) {: Mendefinisikan metode utama (main method) dari kelas SubProgram.
10. int a, b;: Mendeklarasikan dua variabel bertipe integer a dan b.
11. Scanner masukan = new Scanner(System.in);: Membuat objek Scanner untuk membaca masukan dari pengguna.
12. a = masukan.nextInt(); b = masukan.nextInt();: Membaca dua bilangan yang dimasukkan oleh pengguna.
13. System.out.println("Ke dua bilangan : a = " + a + " b = " + b);: Mencetak dua bilangan yang dimasukkan oleh pengguna.
14. System.out.println("Maksimum = " + (maxab(a, b)));: Memanggil subprogram maxab untuk mencari nilai maksimum antara dua bilangan yang dimasukkan oleh pengguna.
15. System.out.println("Tukar kedua bilangan..."); tukar(a, b);: Memanggil subprogram tukar untuk menukar nilai kedua bilangan yang dimasukkan oleh pengguna dan mencetak hasil pertukaran.
16. }: Kurung penutup untuk metode utama (main method).

**19.Tempair**

1. import java.util.Scanner;: Baris ini mengimpor kelas Scanner dari paket java.util, yang digunakan untuk membaca masukan dari pengguna.
2. public class Tempair {: Mendeklarasikan kelas publik dengan nama Tempair.
3. public static void main(String[] args) {: Ini adalah metode utama (main method) dari kelas Tempair, yang merupakan titik masuk (entry point) untuk program. Metode ini dideklarasikan sebagai publik (public), statik (static), dan menerima argumen dalam bentuk array dari string.
4. int T;: Mendeklarasikan sebuah variabel bertipe data integer dengan nama T, yang akan digunakan untuk menyimpan nilai temperatur yang dimasukkan oleh pengguna.
5. Scanner masukan = new Scanner(System.in);: Membuat objek Scanner dengan nama masukan untuk membaca masukan dari pengguna melalui System.in.
6. System.out.print("Contoh IF tiga kasus \n");: Mencetak pesan "Contoh IF tiga kasus" ke konsol.
7. System.out.print("Temperatur (der. C) = ");: Mencetak pesan permintaan kepada pengguna untuk memasukkan nilai temperatur dalam derajat Celsius.
8. T = masukan.nextInt();: Membaca nilai temperatur yang dimasukkan oleh pengguna dan menyimpannya dalam variabel T.
9. if (T < 0) {: Memulai blok pernyataan if, yang akan dieksekusi jika nilai temperatur (T) yang dimasukkan oleh pengguna kurang dari 0.
10. System.out.print("Wujud air beku \n" + T);: Jika nilai temperatur kurang dari 0, program akan mencetak pesan "Wujud air beku" diikuti oleh nilai temperatur tersebut.
11. } else if ((0 <= T) && (T <= 100)) {: Jika nilai temperatur tidak kurang dari 0, maka dilakukan pemeriksaan pada blok else if. Pada blok ini, program akan mengecek apakah nilai temperatur berada di antara 0 dan 100.
12. System.out.print("Wujud air cair \n" + T);: Jika nilai temperatur berada di antara 0 dan 100, program akan mencetak pesan "Wujud air cair" diikuti oleh nilai temperatur tersebut.
13. } else if (T > 100) {: Jika nilai temperatur tidak berada dalam rentang 0 hingga 100, maka program akan mengecek apakah nilai temperatur lebih dari 100.
14. System.out.print("Wujud air uap/gas \n" + T);: Jika nilai temperatur lebih dari 100, program akan mencetak pesan "Wujud air uap/gas" diikuti oleh nilai temperatur tersebut.
15. };: Kurung penutup untuk blok else if.
16. }: Kurung penutup untuk metode utama (main method).