**Kuis Exception Handling**

1. Perhatikan codingan di bawah ini

class Main {

   public static void main(String args[]) {

      try {

         throw 10;

      }

      catch(int e) {

         System.out.println("Saya mendapat exception " + e);

      }

  }

}

Tentukan input yang akan muncul…

1. Saya mendapat exception 10
2. Saya mendapat exception 0
3. Compiler error
4. Perhatikan codingan di bawah ini

class Test extends Exception { }

class Main {

   public static void main(String args[]) {

      try {

         throw new Test();

      }

      catch(Test t) {

         System.out.println("Saya mendapatkan exception test");

      }

      finally {

         System.out.println("Di dalam finally ");

      }

  }

}

Tentukan input yang akan muncul…

1. Saya mendapatkan exception test

Di dalam finally

1. Saya mendapatkan exception test
2. Di dalam finally
3. Compiler error
4. Perhatikan codingan di bawah ini

class Main {

   public static void main(String args[]) {

      int x = 0;

      int y = 10;

      int z = y/x;

  }

}

Apa yang akan terjadi..

1. Compiler error
2. Compiler berjalan dengan baik
3. Compiler berjalan baik tapi melemparkan ArithmeticException
4. Perhatikan codingan di bawah ini

class Base extends Exception {}

class Derived extends Base  {}

public class Main {

  public static void main(String args[]) {

   // some other stuff

   try {

       // Some monitored code

       throw new Derived();

    }

    catch(Base b)     {

       System.out.println("Menangkap base class exception");

    }

    catch(Derived d)  {

       System.out.println("Menangkap derived class exception");

    }

  }

}

Tentukan output yang akan muncul…

1. Menangkap base class exception
2. Menangkap derived class exception
3. Compiler error karena derived tidak throwable
4. Compiler error karena base class exception tertangkap terlebih dahulu sebelum derived class
5. Perhatikan codingan berikut

class Test

{

    public static void main (String[] args)

    {

        try

        {

            int a = 0;

            System.out.println ("a = " + a + "\n");

            int b = 20 / a;

            System.out.println ("b = " + b);

        }

        catch(ArithmeticException e)

        {

            System.out.println ("Error dibagi dengan 0");

        }

        finally

        {

            System.out.println ("Di dalam finally");

        }

    }

}

Tentukan output yang akan muncul…

1. Compile error
2. Error dibagi dengan 0
3. a=0

Error dibagi dengan 0

Di dalam finally

1. a=0
2. Di dalam finally
3. Perhatikan codingan berikut

class Test

{

    public static void main(String[] args)

    {

        try

        {

            int a[]= {1, 2, 3, 4};

            for (int i = 1; i <= 4; i++)

            {

                System.out.println ("a[" + i + "]=" + a[i] + "\n");

            }

        }

        catch (Exception e)

        {

            System.out.println ("error = " + e);

        }

        catch (ArrayIndexOutOfBoundsException e)

        {

            System.out.println ("ArrayIndexOutOfBoundsException");

        }

    }

}

Apa yang terjadi?

1. Compiler Error
2. Run Time Error
3. ArrayIndexOutOfBoundsException
4. Error Code tercetak
5. Array tercetak
6. Perhatikan codingan berikut

class Test

{

    String str = "a";

    void A()

    {

        try

        {

            str +="b";

            B();

        }

        catch (Exception e)

        {

            str += "c";

        }

    }

    void B() throws Exception

    {

        try

        {

            str += "d";

            C();

        }

        catch(Exception e)

        {

            throw new Exception();

        }

        finally

        {

            str += "e";

        }

        str += "f";

    }

    void C() throws Exception

    {

        throw new Exception();

    }

    void display()

    {

        System.out.println(str);

    }

    public static void main(String[] args)

    {

        Test object = new Test();

        object.A();

        object.display();

    }

}

Tentukan output yang akan muncul…

1. abdef
2. abdec
3. abdefc
4. Perhatikan codingan berikut

class Test

{   int count = 0;

    void A() throws Exception

    {

        try

        {

            count++;

            try

            {

                count++;

                try

                {

                    count++;

                    throw new Exception();

                }

                catch(Exception ex)

                {

                    count++;

                    throw new Exception();

                }

            }

            catch(Exception ex)

            {

                count++;

            }

        }

        catch(Exception ex)

        {

            count++;

        }

    }

    void display()

    {

        System.out.println(count);

    }

    public static void main(String[] args) throws Exception

    {

        Test obj = new Test();

        obj.A();

        obj.display();

    }

}

Tentukan output yang akan muncul…

1. 4
2. 5
3. 6
4. Compile Error