



Pengajar:

- Lim Yohanes Stefanus, Ph.D.
- Suryana Setiawan, Ph.D.
- Amril Syalim, Ph.D.
- Dr. Fariz Darari
- Ardhi Putra Pratama Hartono, M.Sc.
- Dipta Tanaya, M.Kom.
- Raja Damanik, M.Sc.

PERNYATAAN KESANGGUPAN MENAATI TATA TERTIB UJIAN

“Saya telah membaca dan memahami ketentuan tata tertib berikut ini, serta menyatakan bahwa jawaban ujian ini adalah hasil pekerjaan saya sendiri. Saya menyetujui jika melakukan pelanggaran atas ketentuan tersebut, saya bersedia diproses sesuai ketentuan yang berlaku (SK DGB UI No.1 Tahun 2014) dengan sanksi maksimal **nilai akhir E.**”

Nama & Tanda-tangan:

Kelas:

Nomor Pokok Mahasiswa:

--

--	--	--	--	--	--	--	--	--	--	--

TATA TERTIB UJIAN

- Semua alat komunikasi elektronik dalam kondisi non-aktif (dimatikan), dimasukkan ke dalam tas dan diletakkan pada tempat yang telah disediakan.
- Peralatan ujian yang boleh dibawa adalah alat tulis dan yang diperbolehkan sesuai sifat ujian.
- Peserta ujian menempati tempat duduk yang telah ditentukan.
- Peserta ujian menuliskan nama dan NPM pada setiap lembar jawaban ujian.
- Peserta mulai membuka soal dan mengerjakan ketika pengawas mengatakan ujian dimulai dan berhenti bekerja (meletakkan alat tulis) ketika pengawas mengatakan waktu habis.
- Peserta tidak berkomunikasi dalam bentuk apa pun dengan peserta lain selama berada di ruang ujian, termasuk pinjam meminjam alat tulis, serta tidak memberi atau menerima bantuan dari siapapun selama ujian berlangsung.
- Peserta yang meninggalkan ruang ujian dianggap selesai mengerjakan. Jika karena kondisi medis khusus tidak bisa memenuhi ketentuan ini, peserta wajib melaporkan kepada pengawas sebelum ujian dimulai.
- Setelah selesai mengerjakan atau setelah waktu habis, peserta segera meninggalkan berkas soal dan lembar jawaban ujian di meja masing-masing, mengambil tas dan segera keluar tanpa mengganggu peserta lain serta tanpa berkomunikasi dengan peserta lain.
- Jawaban ujian ini tidak akan dinilai jika pernyataan di atas ini tidak ditandatangani.

Informasi Tambahan

- Tuliskan nama, NPM, kelas, dan nomor meja di **setiap halaman soal**.
- Ujian bersifat buka catatan yang ditulis dengan tangan pada satu lembar kertas A4.
- Tersedia 16 soal **Pilihan Jamak**, 8 soal **Isian Pendek**, dan 4 soal **Esai**.
- Nilai maksimal yang dapat diperoleh adalah **112 poin**.

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

Pilihan Jamak

Pilih jawaban yang tepat untuk setiap soal dan tuliskan pilihan Anda pada tabel berikut. Setiap soal pilihan jamak berbobot 2 poin.

No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Jwb																

1. Manakah *command* yang tepat untuk mengeksekusi *bytecode* `WelcomeToJava.class`?

- a. `javac WelcomeToJava`
- b. `javac WelcomeToJava.java`
- c. `java WelcomeToJava`
- d. `java WelcomeToJava.class`
- e. `java WelcomeToJava.java`

2. Manakah *statement* tentang deklarasi dan *assignment* variabel yang tidak menyebabkan error?

- a. `int 1 = 2;`
- b. `String _a = "a";`
- c. `String args[] = "b";`
- d. `boolean a = FALSE;`
- e. `int[] arr = [1];`

3. Apakah yang akan dicetak oleh potongan program berikut?

```
int a = 3;
int b = a;
System.out.println(a/b + " == " + b/3.0);
```

- a. `1 == 1.0`
- b. `1 == 1`
- c. `1.0 == 1.0`
- d. `true`
- e. `false`

4. Apakah yang akan dicetak oleh potongan program berikut?

```
boolean a = false || (!false && true);
boolean b = !(true && (false || !true));
System.out.println(b + ":" + a);
```

- a. `true:true`
- b. `true:false`
- c. `false:true`
- d. `false:false`
- e. tidak ada yang dicetak karena ada error

5. Apakah yang akan dicetak oleh potongan program berikut?

```
int val;
String s = "DDP";
for(int i = 0; i < 3; i++) {
    s = i % 3 == 1 ? s.toUpperCase() : s.toLowerCase();
}
s.toUpperCase();
switch(s) {
    case "DDP":
        val = 1; break;
    case "ddP":
        val = 2; break;
    case "DDp":
        val = 3;
    default:
        val = 4; break;
}
System.out.println(val);
```

- a. 1
- b. 2
- c. 3
- d. 4
- e. tidak ada yang dicetak karena ada error

6. Manakah *statement* yang akan menyebabkan *error*?

- a. `int a[] = new int[4];`
- b. `int[] a = new int[0];`
- c. `int a[][] = new int[4];`
- d. `int[][] a = new int[0][4];`
- e. `int[] a[] = new int[4][4];`

7. Manakah *statement* yang legal?

- a. `int[] a = { {} };`
- b. `int[] a = { {1} };`
- c. `int[][] a = {1, 2, 3, 4};`
- d. `int[][] a = { 1, {1, 2}, {3, 4}};`
- e. `int[][] a = { {}, {1, 2}, {3, 4}};`

8. Perhatikan potongan kode berikut.

```
ArrayList<Integer> list = new ArrayList<Integer>();
list.add(5); list.add(0, 4); list.add(0, 3); list.add(0, 2);
list.add(0, 1);
```

Manakah isi list yang paling tepat setelah potongan kode tersebut dieksekusi?

- a. [5, 4, 3, 2, 1]
- b. [5, 0, 0, 0, 0]
- c. [1, 2, 3, 4, 5]
- d. [0, 1, 2, 3, 4]
- e. [1]

9. Apa yang akan terjadi dengan kode di bawah ini bila di-*compile* dan dieksekusi (apabila berhasil dikompilasi)?

```
public class Question09 {
    public static void main(String[] args) {
        ArrayList ageList = new ArrayList();
        ageList.add(new Integer("17"));
        ageList.add(50);

        Integer sum = new Integer(0);
        for (Integer age : ageList) {
            sum = sum + age;
        }
        System.out.println(sum);
    }
}
```

- a. Tidak dapat dikompilasi karena parameter dalam new Integer tidak boleh bertipe String
- b. Tidak dapat dikompilasi karena tidak bisa mengkonversi tipe data pada ArrayList ke Integer
- c. Tidak dapat dikompilasi karena tidak menspesifikasikan tipe pada deklarasi ArrayList
- d. Bisa dikompilasi, namun error saat dijalankan
- e. Bisa dikompilasi dan bisa dijalankan tanpa masalah

10. Jika parameter int x diasumsikan bernilai lebih dari 1, isian yang tepat untuk melengkapi *method* berikut agar *method* sum mengembalikan hasil jumlah $1+2+\dots+x$ adalah ...

```
public static int sum(int x){
    if(x == 1) return 1;
    else return _____;
}
```

- a. $x + \text{sum}(x-1)$
- b. $x + \text{sum}(x)$
- c. $\text{sum}(x-1)$
- d. $\text{sum}(x+1)$
- e. $\text{sum}(x)$

11. Diberikan dua *method* berikut, maka $f(5)$ akan mengembalikan nilai

```
public static int f(int n) {
    if(n <= 1)
        return 1;
    return n + g(n+1);
}
public static int g(int n) {
    if(n < 1)
        return 1;
    return n * f(n-2);
}
```

- a. 5
- b. 7
- c. 20
- d. 23
- e. tidak ada yang dicetak karena ada error

12. Mengacu pada *method* di nomor 11, maka $g(4)$ akan mengembalikan nilai

- a. 3
- b. 7
- c. 20
- d. 23
- e. tidak ada yang dicetak karena ada error

13. Class Penduduk berikut ini memiliki beberapa *method*

```
class Penduduk{
    String nama;
    int umur;

    public void setNama(String s){nama = s;}
    public void setUmur(int i){umur = i;}
    public int getUmur(){ return umur;}
    public String getKategoriUmur(int i){
        if(i < 20) return "remaja";
        else if(i < 45) return "dewasa";
        else if(i < 60) return "pra lansia";
        else return "lansia";
    }
    public String getKategoriUmur(){
        return getKategoriUmur(umur);
    }
}
```

Dari *method* tersebut, mana yang paling tepat dibuat sebagai *static method*?

- a. setNama(String s)
- b. setUmur(int i)
- c. getUmur()
- d. getKategoriUmur(int i)
- e. getKategoriUmur()

14. Apakah keluaran dari potongan kode berikut ketika dikompilasi dan dijalankan?

```
public class Question14{
    static int i=1;
    public static void main(String[] args){
        int j = i;
        j = 3;
        Question14 q14 = new Question14();
        System.out.print(Question14.i + ":" + q14.i + ":" + q14.j);
    }
}
```

- a. tidak ada yang dicetak karena kesalahan kompilasi
- b. 1 1 1
- c. 1 1 3
- d. 1 3 1
- e. 1 3 3

15. Apakah keluaran dari program berikut?

```
class B {
    int vf;
    static int sf = 10;
    B(int i) {
        vf = i;
        sf += i;
    }
    B() {
        this(10);
    }
    public String toString(){
        return "vf="+ this.vf + ",sf=" + this.sf + ".";
    }
}
public class UTS {
    public static void main(String[] args){
        B b1 = new B();
        B b2 = new B(10);
        b1 = b2;
        b2.vf = 10;
        b2 = new B(100);
        System.out.println(b1+" "+b2);
    }
}
```

- a. vf=10,sf=130. vf=100,sf=130.
- b. vf=10,sf=130. vf=100,sf=130.
- c. vf=10,sf=130. vf=100,sf=130.
- d. vf=10,sf=130. vf=100,sf=130.
- e. vf=10,sf=130. vf=100,sf=130.

16. Apa keluaran program berikut?

```
String s1 = new String("DDP");
String s2 = new String("DDP");
String s3 = s2;

System.out.print(s1==s2);
System.out.print(s2==s3);
System.out.print(s1.equals(s2));
```

- a. truetruetrue
- b. truetruefalse
- c. falsetruefalse
- d. falsetruetrue
- e. semua pilihan a, b, c, d salah.

ISIAN PENDEK

Jawablah dengan singkat dan jelas pada ruang yang disediakan. Setiap soal isian pendek berbobot 5 poin.

1. Perhatikan potongan kode berikut.

```
String s = "fakultas ilmu komputer";
int i = 1;
while(i < s.length()){
    System.out.print(s.charAt(i));
    i += 2;
}
```

- a. Keluaran dari program tersebut adalah _____
- b. Ubah potongan kode tersebut ke bentuk for loop yang setara.

```
String s = "fakultas ilmu komputer";
for(_____; _____; _____){
    System.out.print(s.charAt(i));
}
```

2. Keluaran dari program berikut adalah (atau apakah **ada error**): _____

```
int n = 0;
for(int i = 0; i < 5; i += 5) {
    i -= 4;
    for(int j = 0; j < 10; j += 10) {
        j -= 2;
        n++;
    }
}
System.out.println(n);
```

3. Perhatikan potongan program berikut.

```
public static int mystery(String str, char x) {
    int result = 0;
    if (str.length() > 0){
        result = mystery( str.substring(1), x )
            + ( (str.charAt(0) == x) ? 1 : 0 );
    }
    return result;
}
```

- a. Keluaran dari `mystery("ada apa?", a)` adalah _____
- b. Apa yang dilakukan *method* `mystery`?

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

4. Lengkapi kode berikut sehingga akan menghasilkan pasangan masukan dan keluaran sebagai berikut:

masukan	1	2	3	4	5	dst
keluaran	1	3	4	7	9	dst

```
public static int f(int n) {
    if(n <= 1)
        return 1;
    return n + f(_____);
}
```

5. Perhatikan potongan kode berikut. Apa keluaran program berikut?

```
public class UTS{
    public static void main(String[] args){
        int[] a = {1,2,3};
        int[] b = a;
        b[0] = a[1];
        printArray(a); // baris 1
        b = doSomething(a);
        printArray(a); // baris 2
        printArray(b); // baris 3
    }
    public static int[] doSomething(int[] a){
        int[] b = new int[3];
        a = b;
        return b;
    }
    public static void printArray(int[] a){
        for (int i=0; i<a.length; i++){
            System.out.print(a[i]+" ");
        }
        System.out.println();
    }
}
```

Keluaran pada baris 1: _____

Keluaran pada baris 2: _____

Keluaran pada baris 3: _____

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

6. Lengkapi implementasi *method* `toArrayList(T[] arr)` yang menerima sebuah array dan mengembalikan `ArrayList<T>` dengan elemen yang sama persis dengan `arr` dalam urutan yang sama pula. Tipe data elemen independen terhadap logika implementasi metode ini, sehingga *method* yang diimplementasikan seharusnya bersifat *generic*.

```
public static ____(1)____ ArrayList<T> toArrayList(T[] arr){
    ArrayList<T> list = ____ (2) ____;
    for(int i=0; i<arr.length; i++){
        ____ (3) ____;
    }
    return list;
}
```

Isian untuk (1): _____

Isian untuk (2): _____

Isian untuk (3): _____

7. *Method* `hasDuplicate` berikut menerima sebuah array of int dan mengembalikan nilai true jika mengandung duplikasi. Contoh {1, 2, 3, 2} akan mengembalikan nilai true dan {1, 2, 3} mengembalikan false. Buat implementasinya.

```
public boolean hasDuplicate(int[] a){
    int n = a.length;
    for (int i=0; i<n; i++){
        for (int j=_____; j<n; j++){
            if (_____) {
                return ____;
            }
        }
    }
    return ____;
}
```

8. Perhatikan potongan kode berikut:

```

public class Bolu {
    private boolean sudahPanggang;
    private int tepungTerigu;
    private int kuningTelur;
    private int putihTelur;
    private int gulaPasir;
    private String deskripsi;

    private static int urutanBolu = 0;

    private Bolu() {
        this.sudahPanggang = false;
        this.urutanBolu++;
    }
    private Bolu(int tepungTerigu, int kuningTelur, int putihTelur, int gulaPasir) {
        _____ (1) _____;
        this.tepungTerigu = tepungTerigu;
        this.kuningTelur = kuningTelur;
        this.putihTelur = putihTelur;
        this.gulaPasir = gulaPasir;
        this.deskripsi = "Kue Bolu, Belum Dipanggang";
    }
    public void panggang() {
        if(!isSudahPanggang()) {
            setDeskripsi("Kue Bolu, Sudah Dipanggang");
            setSudahPanggang(true);
        }
        else setDeskripsi("Kue Bolu, Gosong");
    }
    public String cetakBolu() {
        return "Bolu ke-" + _____ (2) _____ + " status : "+deskripsi;
    }
    public boolean isSudahPanggang() {
        return _____ (3) _____;
    }
    public void setSudahPanggang(boolean apapun) {
        sudahPanggang = apapun;
    }
    public void setDeskripsi(String deskripsi) {
        this.deskripsi = deskripsi;
        return;
    }
    public static void main(String[] args) {
        Bolu boluPertama = new Bolu(200, 6, 4, 200);
        System.out.println(boluPertama.cetakBolu());
        boluPertama.panggang();
        System.out.println(boluPertama.cetakBolu());

        Bolu boluKedua = new Bolu(300, 8, 5, 300);
        System.out.println(boluKedua.cetakBolu());
        boluKedua.panggang();
        System.out.println(boluKedua.cetakBolu());

        Bolu boluKetiga = boluPertama;
        System.out.println(boluKetiga.cetakBolu());
        boluKetiga.panggang();
        System.out.println(boluKetiga.cetakBolu());
    }
}

```

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

Lengkapilah class Bolu tersebut sehingga program dapat dijalankan. Masing-masing isian hanya boleh diisi oleh : **alfabet, angka**, dan simbol-simbol berikut (5 simbol) : () _ , dan . .

Keluaran yang diharapkan :

```
Bolu ke-1 status : Kue Bolu, Belum Dipanggang
Bolu ke-1 status : Kue Bolu, Sudah Dipanggang
Bolu ke-2 status : Kue Bolu, Belum Dipanggang
Bolu ke-2 status : Kue Bolu, Sudah Dipanggang
Bolu ke-2 status : Kue Bolu, Sudah Dipanggang
Bolu ke-2 status : Kue Bolu, Gosong
```

Isian untuk (1): _____

Isian untuk (2): _____

Isian untuk (3): _____

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

ESAI

Jawablah setiap soal pada ruang yang disediakan. Setiap soal esai berbobot 10 poin.

Esai 1.

Implementasikan *method* `getMostFreqDigit(int n)` yang menerima satu argumen `int`, dan mengembalikan digit yang paling sering muncul pada `int` tersebut. Asumsikan tidak ada *leading zero* di argumen `int`-nya (misalnya, tidak boleh ada masukan 0010). Apabila ada *tie* (frekuensi tertinggi ada pada lebih dari satu digit angka), maka kembalikan salah satu digit saja (bebas yang mana).

Berikut adalah contoh pemanggilan *method* ini:

Pemanggilan <i>method</i>	Nilai yang dikembalikan
<code>getMostFreqDigit(1222513)</code>	2
<code>getMostFreqDigit(1253)</code>	1

Gunakan halaman balik jika perlu.

Jawaban.

```
public static getMostFreqDigit(int n)
```

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

//Lembar jawaban tambahan

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

Esai 2.

Lengkapilah program di bawah ini yang bertujuan untuk membuat sebuah class bernama `HitungNilai`, yang membaca daftar nilai dari console oleh pengguna di mana setiap nilai harus berupa integer `int` pada rentang 0 hingga 100 (inklusif).

Program mencetak pesan

Masukkan sebuah `int` antara 0 dan 100 (inklusif) atau -1 untuk keluar:
ke console setiap kali sebelum membaca `int` berikutnya. Simpan semua `int` yang dimasukkan oleh pengguna ke dalam sebuah array atau `ArrayList` (yang menurut Anda tepat). Diasumsikan pengguna memasukkan setidaknya satu buah nilai pada rentang 0 hingga 100 dan tidak pernah memasukkan nilai di luar rentang tersebut, kecuali -1 (untuk keluar); namun program diharapkan dapat memproses nilai sebanyak-banyaknya selama pengguna memasukkan nilai.

Jika pengguna memasukkan -1, program berhenti dan akan mengembalikan nilai maksimum dan nilai rata-rata dari semua nilai yang dimasukkan oleh pengguna (selain -1).

Berikut adalah contoh eksekusi program `HitungNilai.java`.

```
Masukkan sebuah int antara 0 dan 100 (inklusif) atau -1 untuk keluar: 10
Masukkan sebuah int antara 0 dan 100 (inklusif) atau -1 untuk keluar: 15
Masukkan sebuah int antara 0 dan 100 (inklusif) atau -1 untuk keluar: -1
Nilai maksimum: 15
Nilai rata-rata: 12.5
```

Program `HitungNilai.java`:

```
import

public class HitungNilai{
    public static void main(String[] args){
        //Deklarasikan array atau arraylist yang Anda gunakan untuk menyimpan nilai

        String pesan = "Masukkan sebuah int antara 0 dan 100 (inklusif)
                        atau -1 untuk keluar: ";

        System.out.print(pesan);
        int grade = sc.nextInt();
        while(grade != -1){

        }
    }
}
```

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

```
int max; //variabel untuk menyimpan nilai maksimum
double avg; //variabel untuk menyimpan rata-rata

//hitung nilai maksimum dan simpan dalam max

//hitung nilai rata-rata di sini dan simpan dalam avg

System.out.println("Nilai maksimum: " + max);
System.out.println("Nilai rata-rata:" + avg);
}
}
```

Apakah program tersebut sudah diimplementasikan dengan baik secara OOP? Berikan alasan singkat (maksimal 30 kata).

Esai 3.

Pada soal ini, Anda diminta membuat implementasi class Atom. Setiap atom memiliki banyak proton, neutron, dan elektron yang berbeda-beda, misalnya atom Hidrogen memiliki satu buah proton, satu buah elektron, dan tanpa neutron.

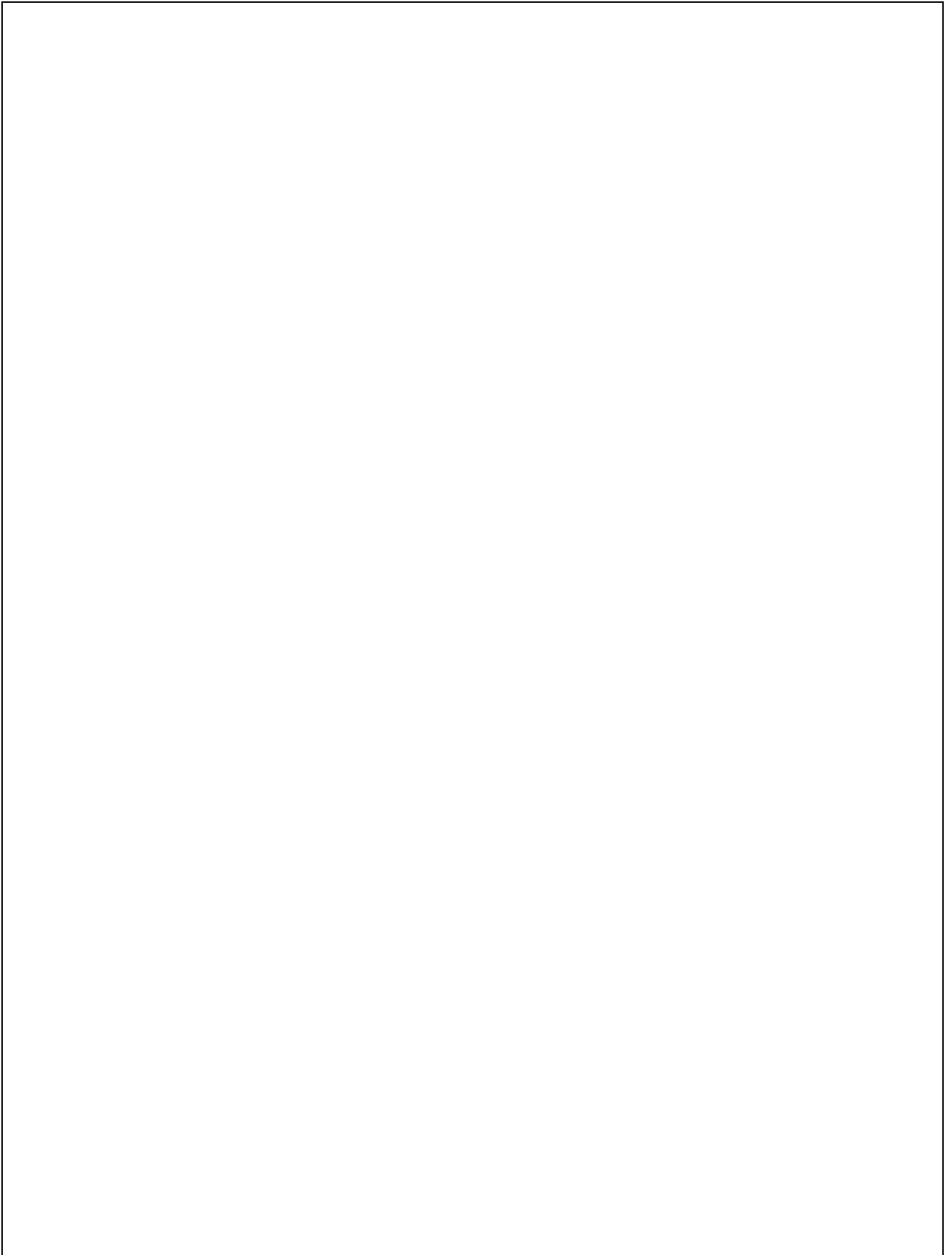


Berikut adalah spesifikasi class Atom:

1. Beberapa *instance variable* yang masing-masing menyimpan banyak proton, neutron, dan elektron. Pilih *access modifier* yang membuat variabel-variabel ini hanya bisa diakses oleh class Atom sendiri.
2. Buat variable untuk mencatat banyak atom yang sudah dibuat oleh class Atom. Pilih *access modifier* yang membuat variabel-variabel ini hanya bisa diakses oleh class Atom sendiri.
3. *Accessor* dan *mutator* untuk variabel pada poin (1) dan *accessor* untuk variabel pada poin (2).
4. Sebuah *constructor* berargumen banyak proton, neutron, dan elektron.
5. Sebuah *constructor* tanpa argumen (*no-arg*) yang membuat atom Hidrogen. Constructor ini memanggil *constructor* berargumen untuk mengisi *instance variable*.
6. Sebuah *method* pada class Atom yang disebut *isIon* yang mengecek apakah suatu atom merupakan ion. Suatu atom merupakan ion jika banyak protonnya tidak sama dengan banyak elektron.
7. Sebuah *method* yang disebut *getAtomicMassNumber* yang akan menghitung dan mengembalikan nomor massa atom. Nomor massa dari sebuah atom dihitung dengan rumus banyak proton ditambah banyak neutron.

Gunakan halaman balik jika perlu.

Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____



Nama: _____ NPM: _____ Kelas: _____ No. Meja: _____

//Lembar jawaban tambahan