DAFTAR PUSTAKA

1. International Union of Pure and Applied Chemistry. "alkanes". Compendium of Chemical Terminology (http://wwwsst.ums.edu.my/data/file/B08ZjmO59pah)
2. Nely Indra Meifiani. 2008. Aplikasi Cayley Tree Dalam Menentukan Banyak Isomer Senyawa Alkana, Yogyakarta: Universitas Negeri Yogyakarta
3. Aldous, Joan M. and Wilson Robin J. 2004. Graphs and Aplication. Springer. Britain.
4. Bollobas, Bela. 1979. Graph Theory An Introductory Course. Springer-Verlag. England.
5. Chartrand, Gary. And Zhang Ping. 2009. Chromatic Graph Theory. CRC Press. USA.
6. Okki Riza Wibisono. 2010, Penerapan Teori Graf Untuk Menghitung Jumlah Isomer Alkana, Bandung: Institut Teknologi Bandung
7. Guillermo Restrepo, José L.Villaveces.2012, *Mathematical Thinking in Chemistry*, HYLE--International Journal for Philosophy of Chemistry, Vol. 18, No.1 (2012), pp. 3-22.
8. Istvan Lukovits, 2004, *Constuctive Enumeration of Chiral Isomer of Alkanes*. Jurnal Isomer, Croatica Chemica Acta Ccacaa77(1–2) 295¿300 (2004) Issn-0011-1643
9. Kevin Ballard, Advisor M. Eicher,Mentor N. Preyer. 2003, *Computerized Isomer Enumeration of the Alkane Series*, Journal of Symbolic.
10. Kristian H Sugiyarto. 2006. Dasar-dasar Kimia Anorganik Transisi. Yogyakarta: Universitas Negeri Yogyakarta.

Lampiran 1. Hasil Pengujian

2. Pengujian kedua

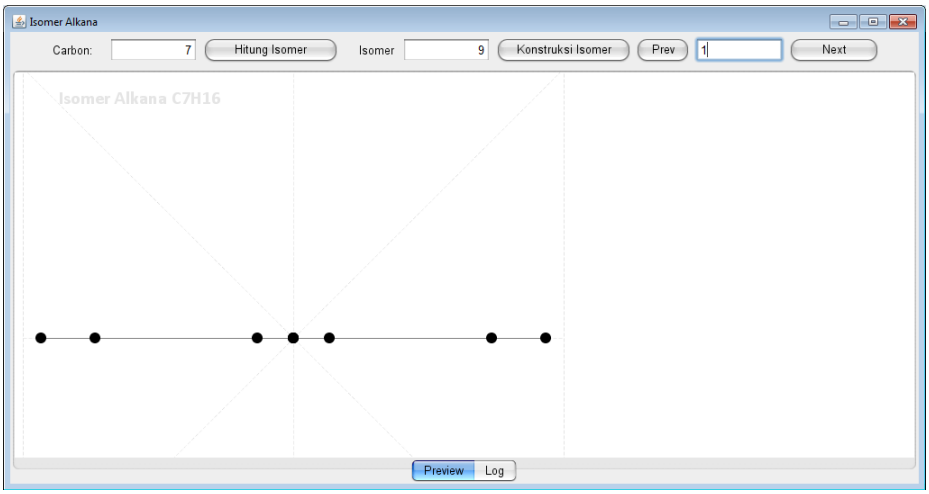
Input: Carbon = 7

Diperoleh output: Jumlah Isomer Alkana Untuk C7H16 adalah: 9

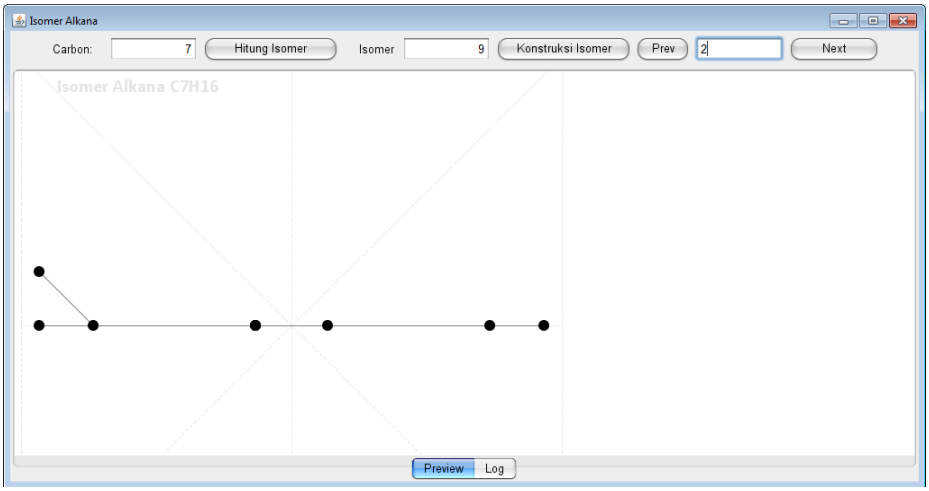
Rumus Struktur Pohon Centered dan Bicentered

- 121111
- 221121
- 2221101
- 13210
- 1222
- 1231
- 141100
- 13111
- 2232

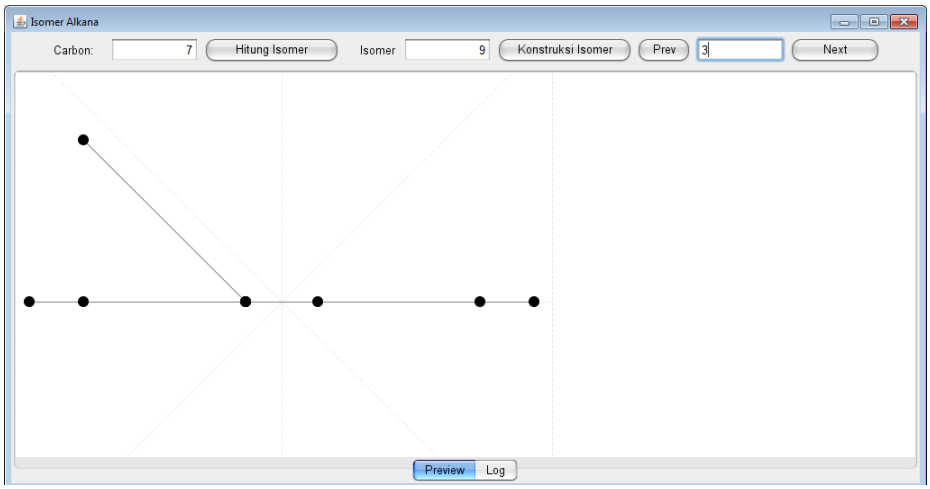
Preview Isomer:



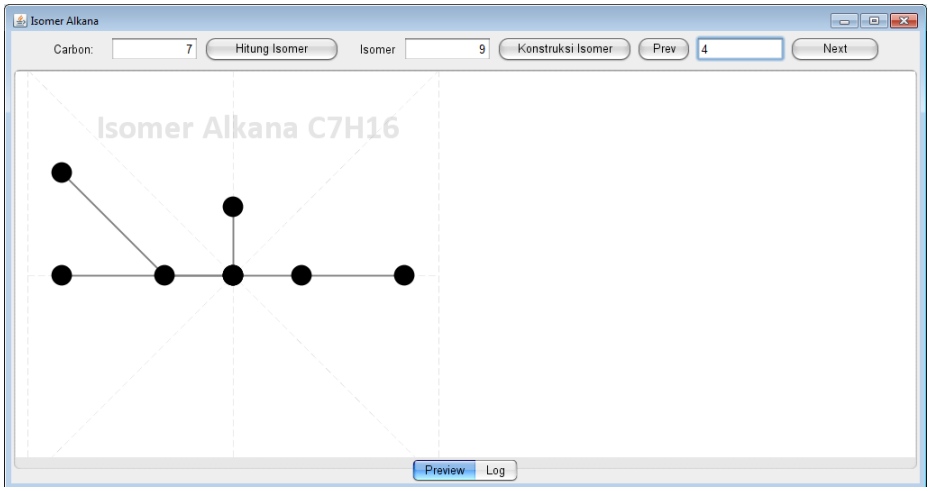
Gambar : Isomer-1 C7H16



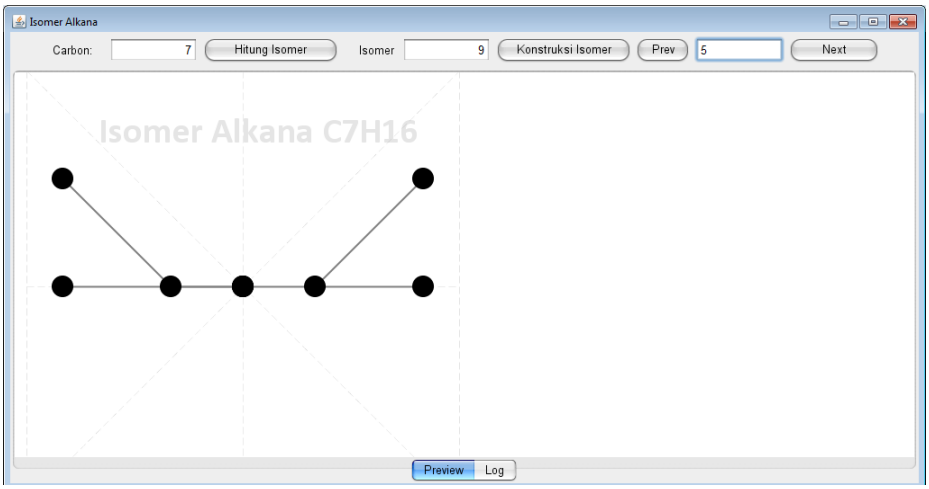
Gambar : Isomer-2 C7H16



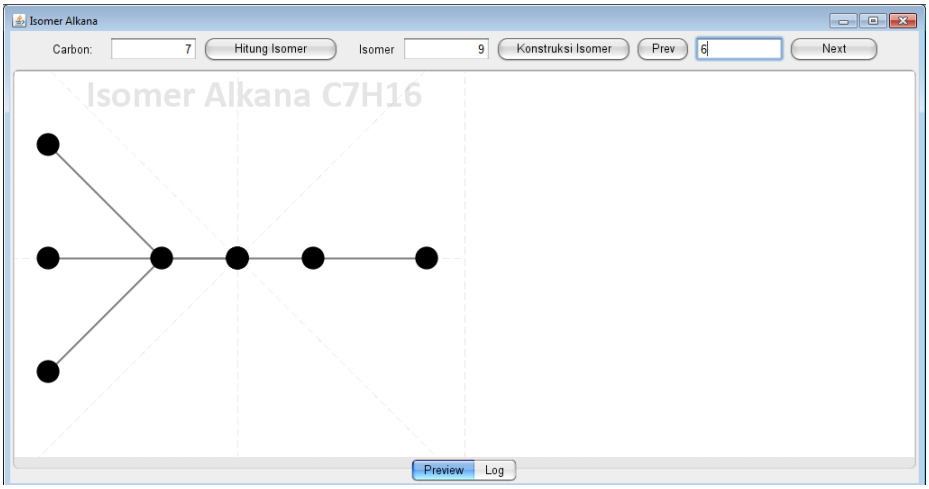
Gambar : Isomer-3 C7H16



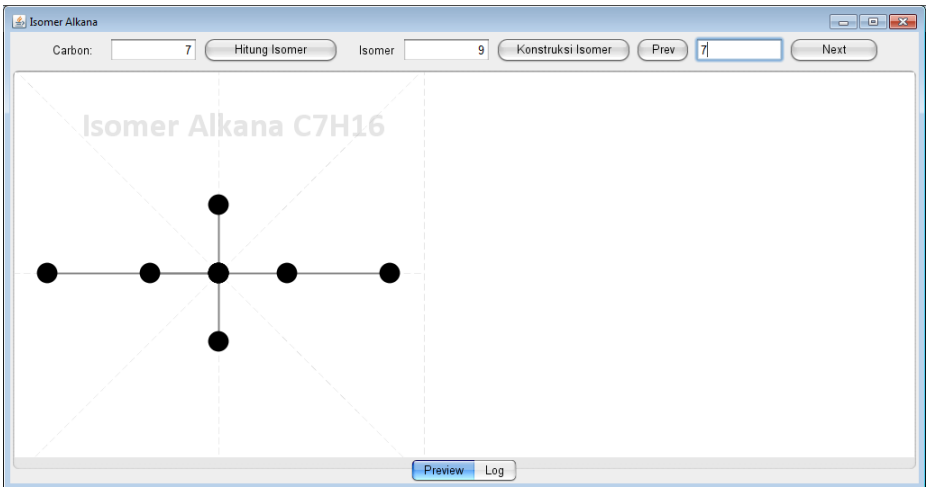
Gambar : Isomer-4 C7H16



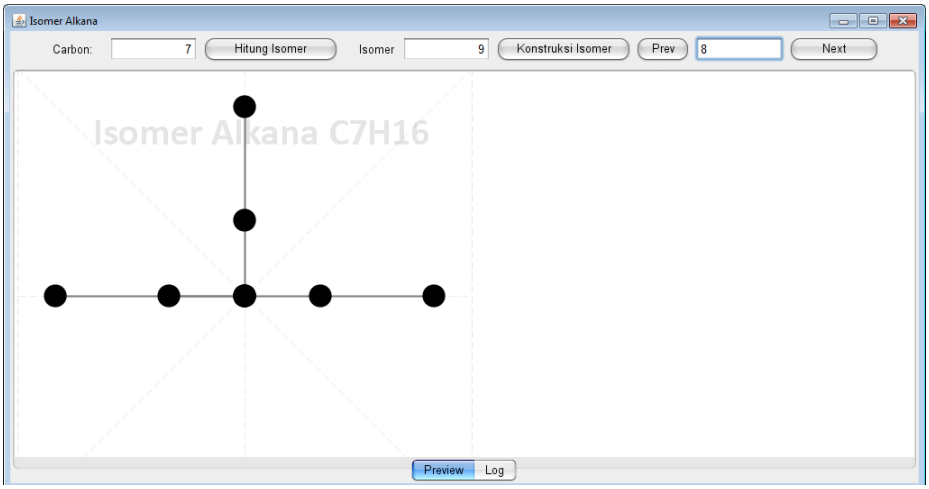
Gambar : Isomer-5 C7H16



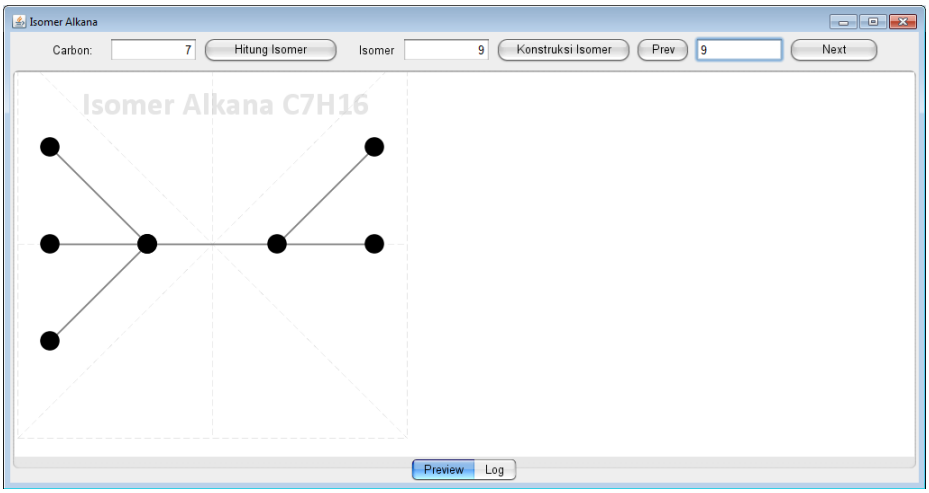
Gambar : Isomer-6 C7H16



Gambar : Isomer-7 C7H16



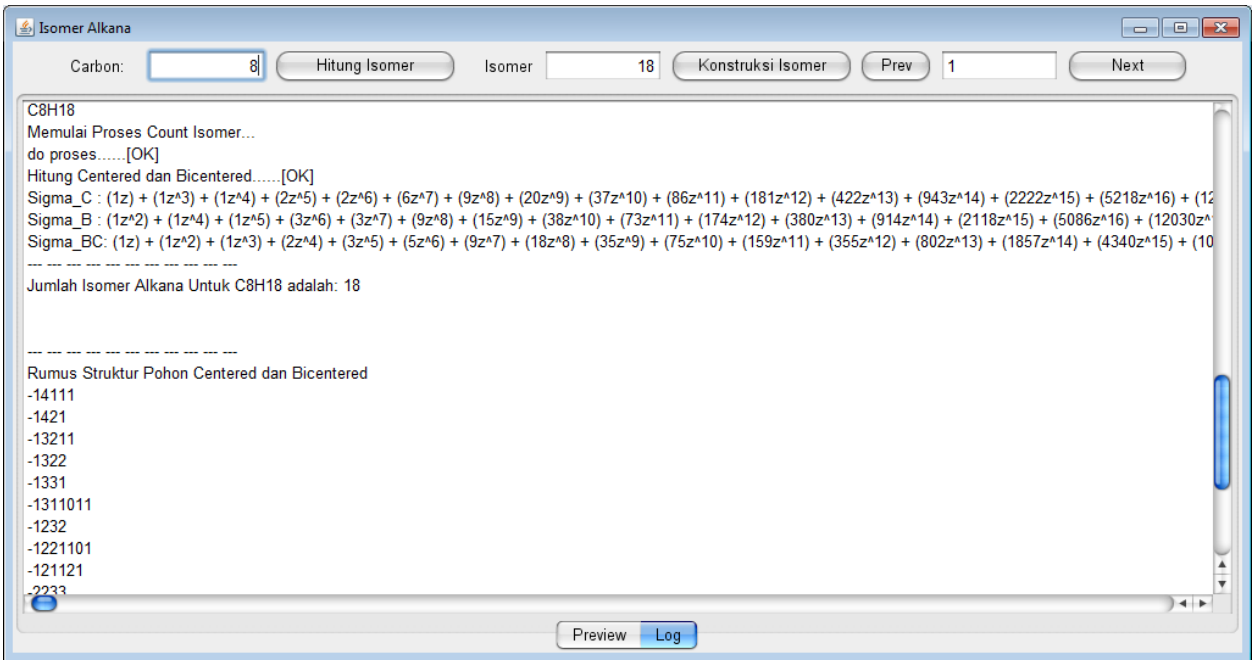
Gambar : Isomer-8 C7H16



Gambar : Isomer-9 C7H16

3. Pengujian ketiga

Input: Carbon = 8



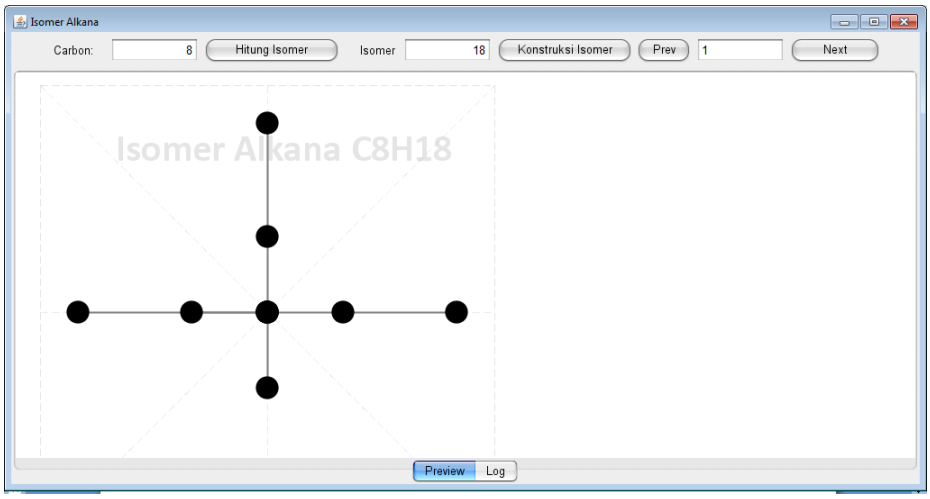
Diperoleh output: Jumlah Isomer Alkana Untuk C8H18 adalah: 18

Rumus Struktur Pohon Centered dan Bicentered

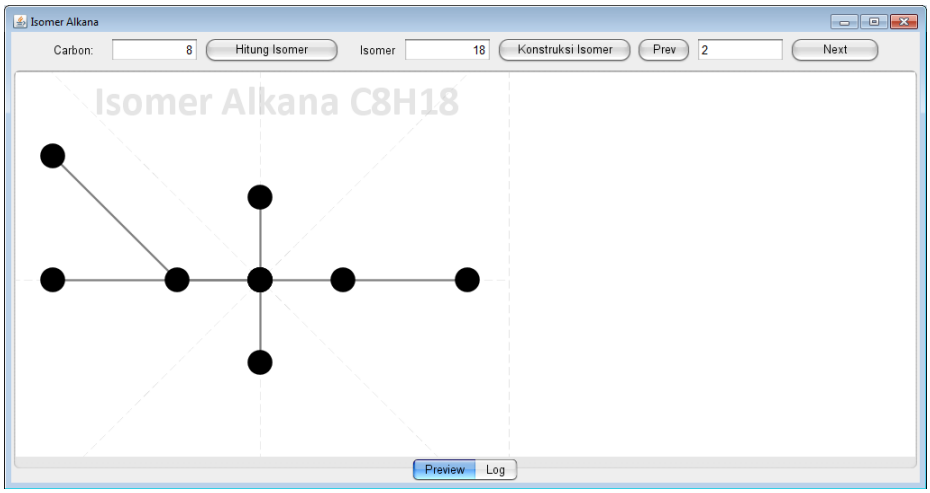
- 14111
- 1421
- 13211

- 1322
- 1331
- 1311011
- 1232
- 1221101
- 121121
- 2233
- 22221010
- 22311001
- 22111111
- 2221111
- 221122
- 2221201
- 2221102
- 221131

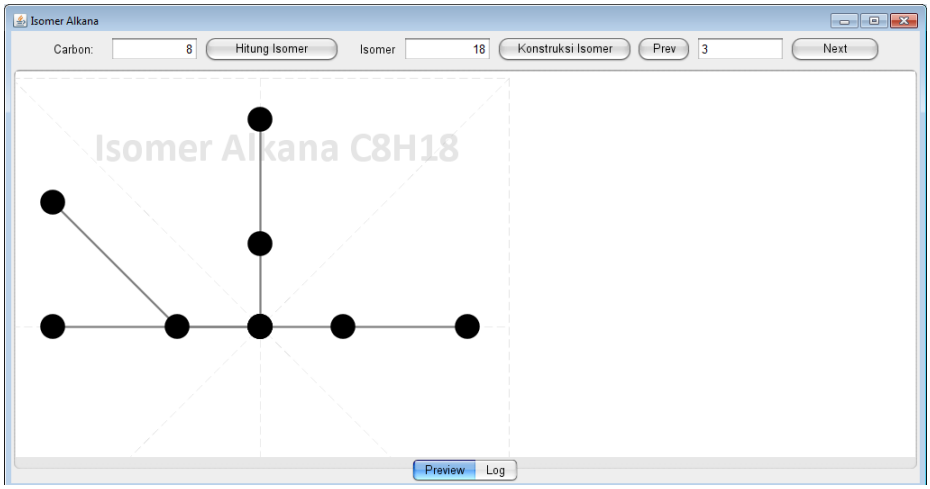
Preview Isomer:



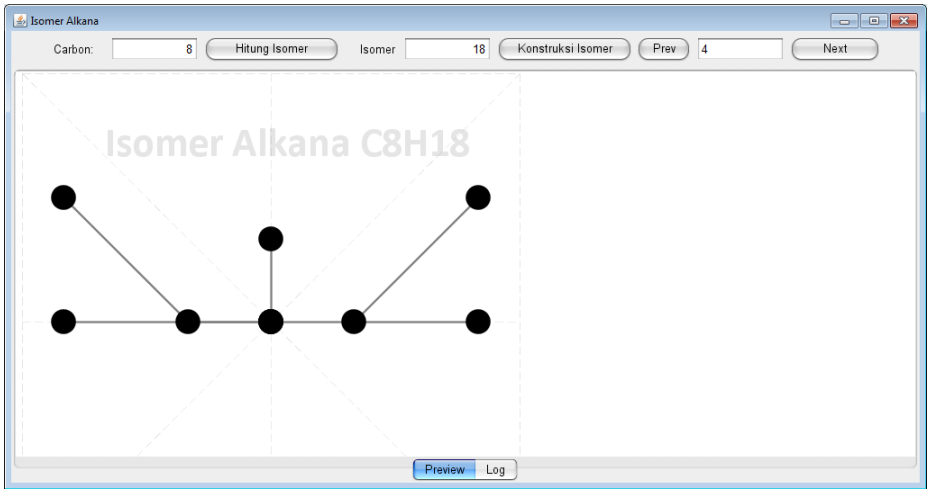
Gambar : Isomer-1 C8H18



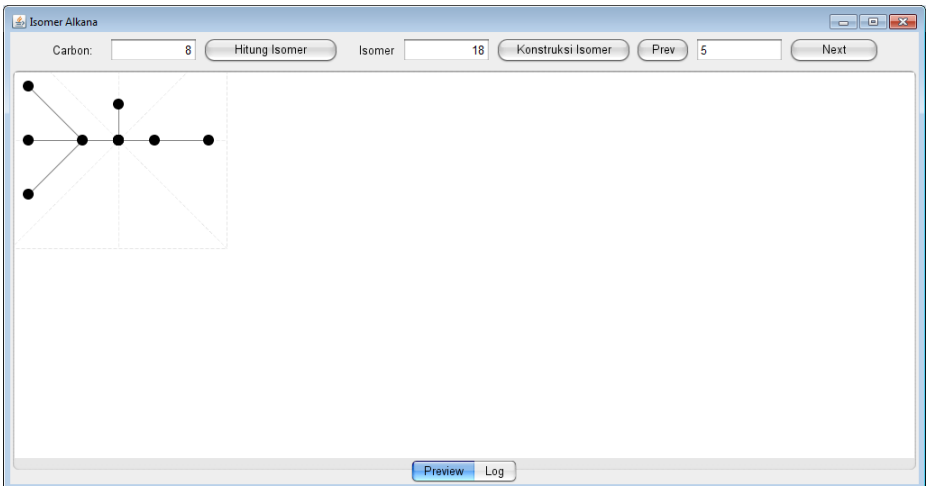
Gambar : Isomer-2 C8H18



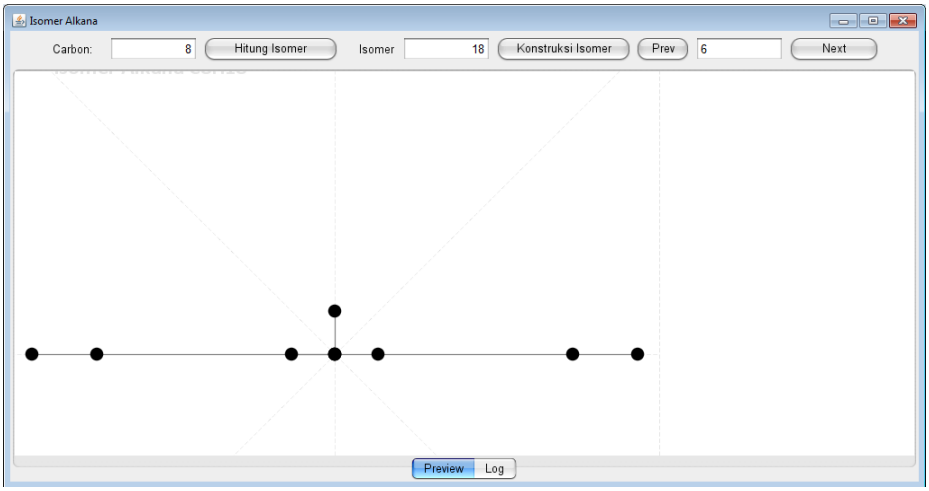
Gambar : Isomer-3 C8H18



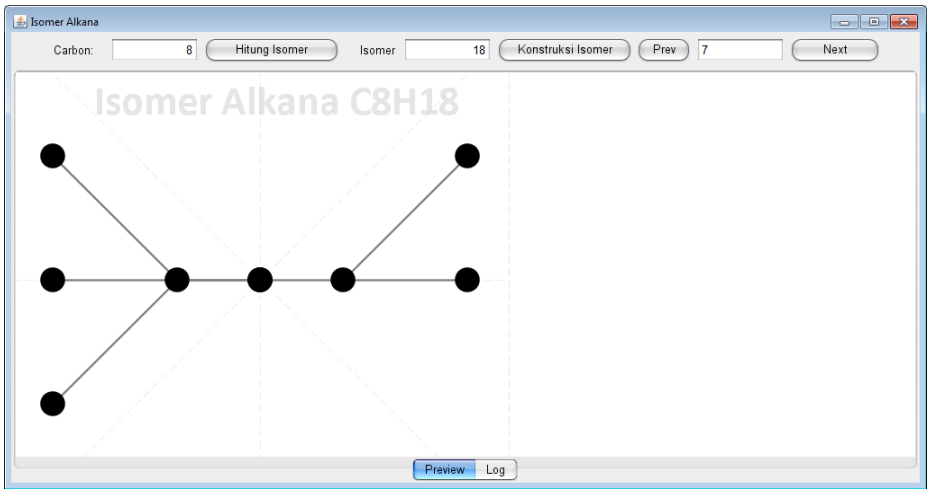
Gambar : Isomer-4 C8H18



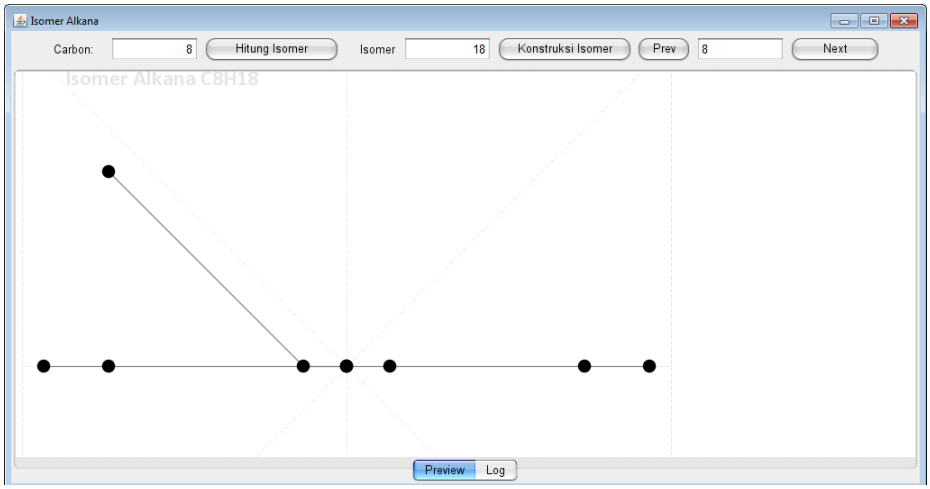
Gambar : Isomer-5 C8H18



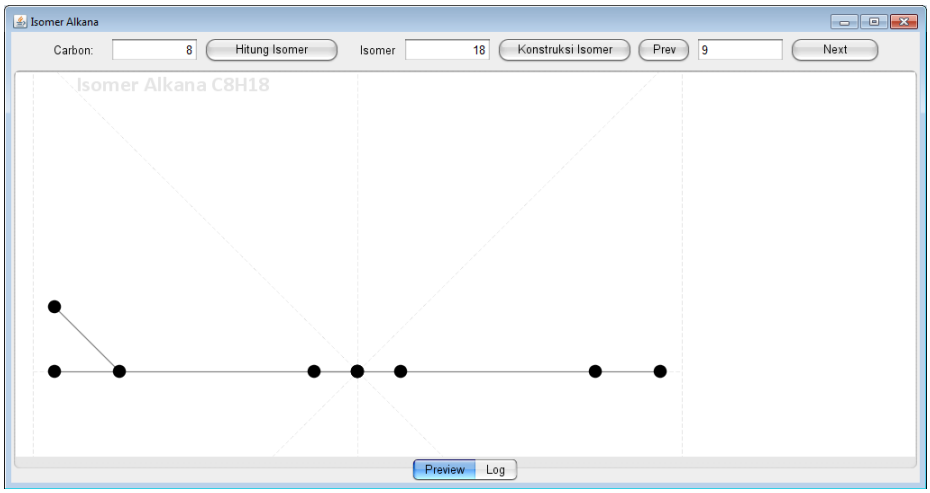
Gambar : Isomer-6 C8H18



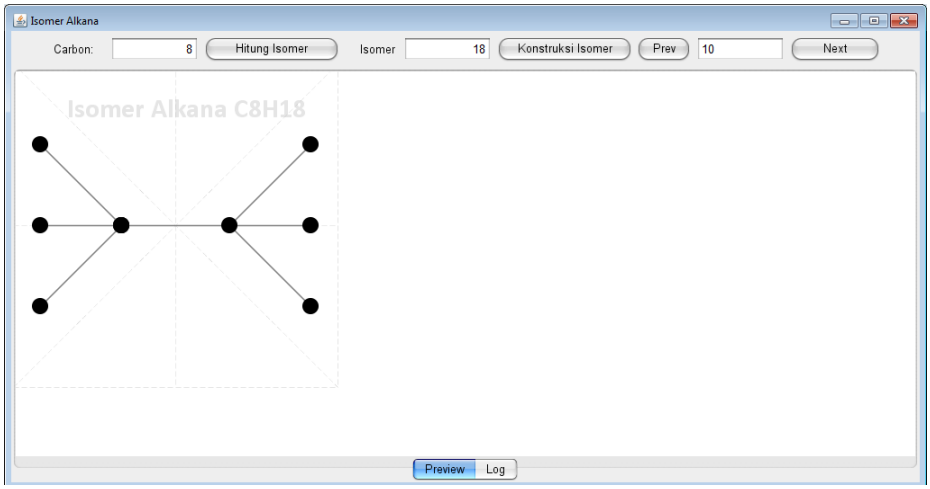
Gambar : Isomer-7 C8H18



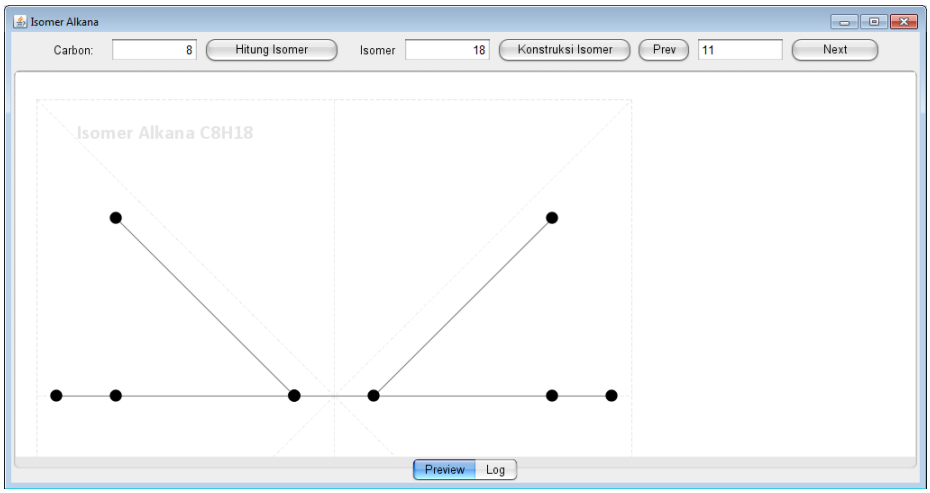
Gambar : Isomer-8 C8H18



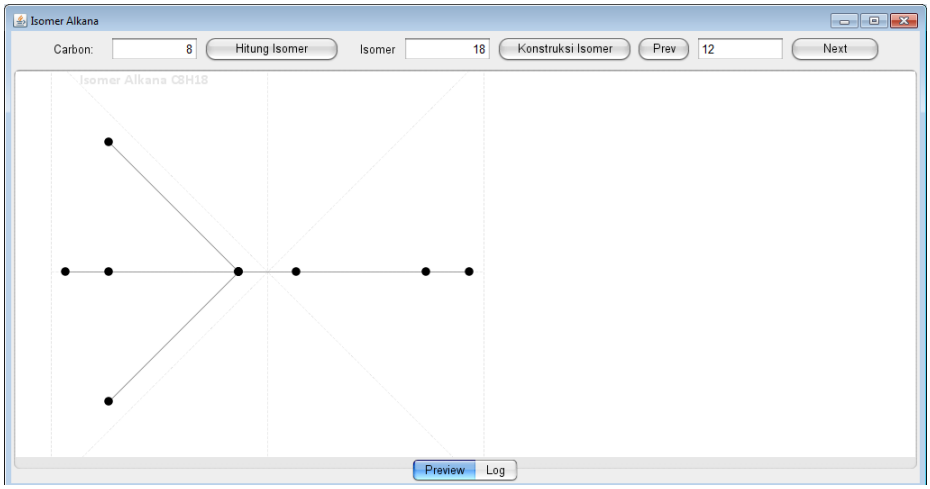
Gambar : Isomer-9 C8H18



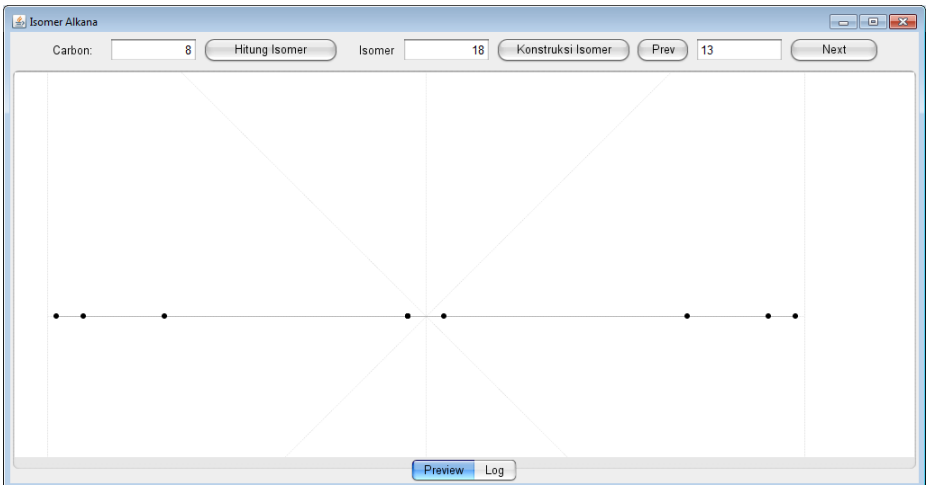
Gambar : Isomer-10 C8H18



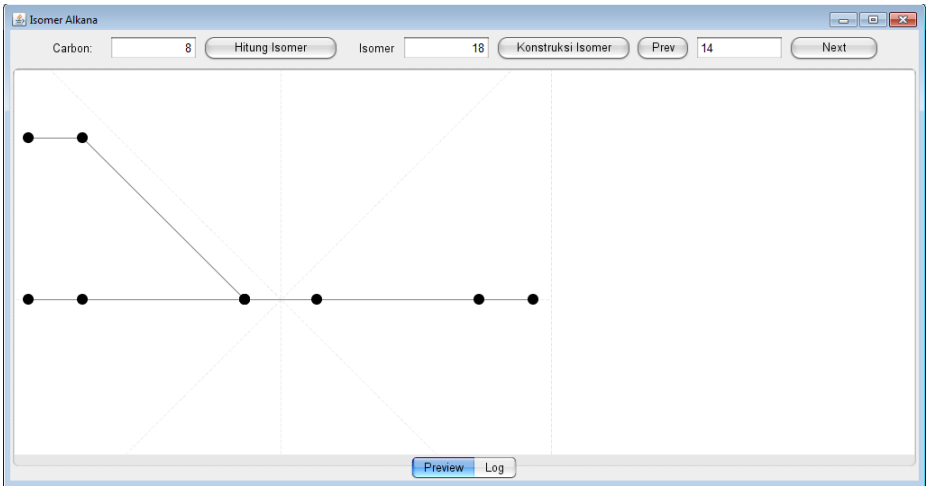
Gambar : Isomer-11 C8H18



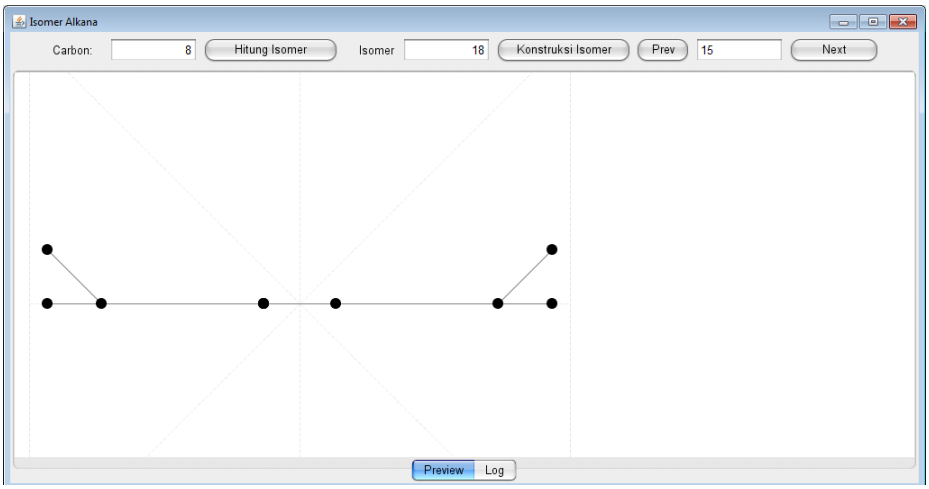
Gambar : Isomer-12 C8H18



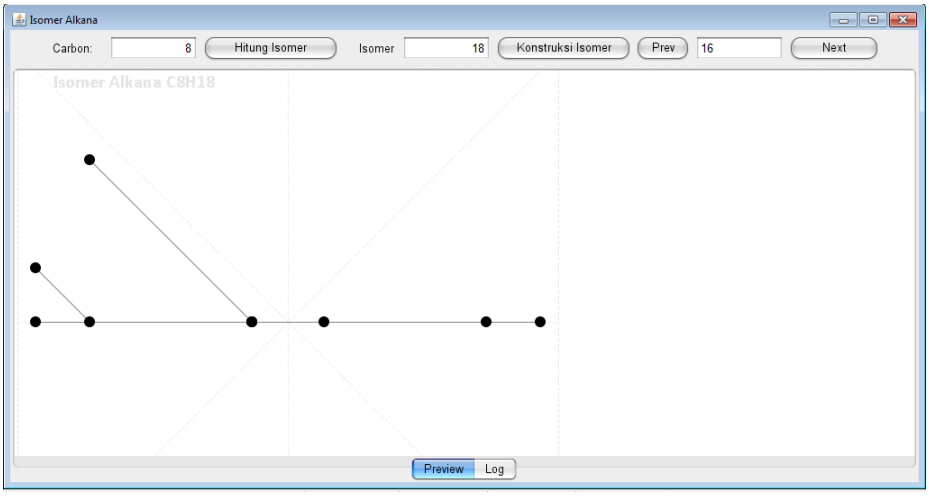
Gambar : Isomer-13 C8H18



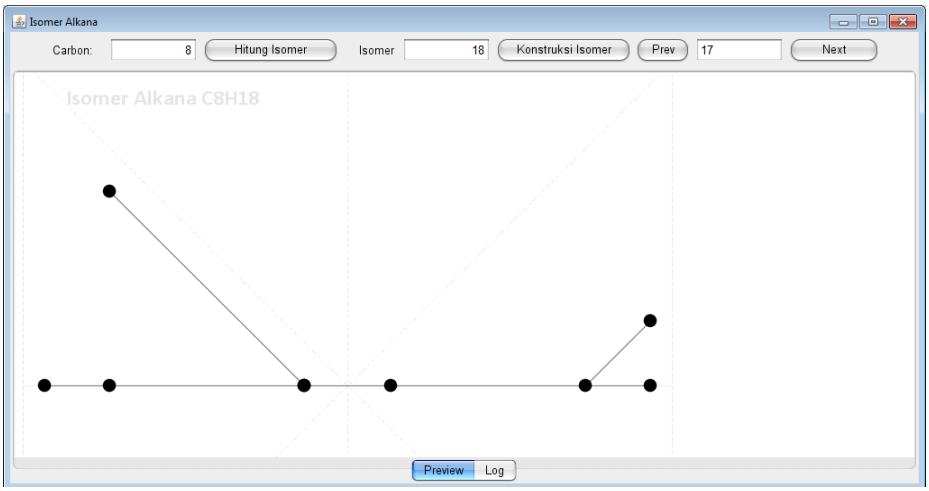
Gambar : Isomer-14 C8H18



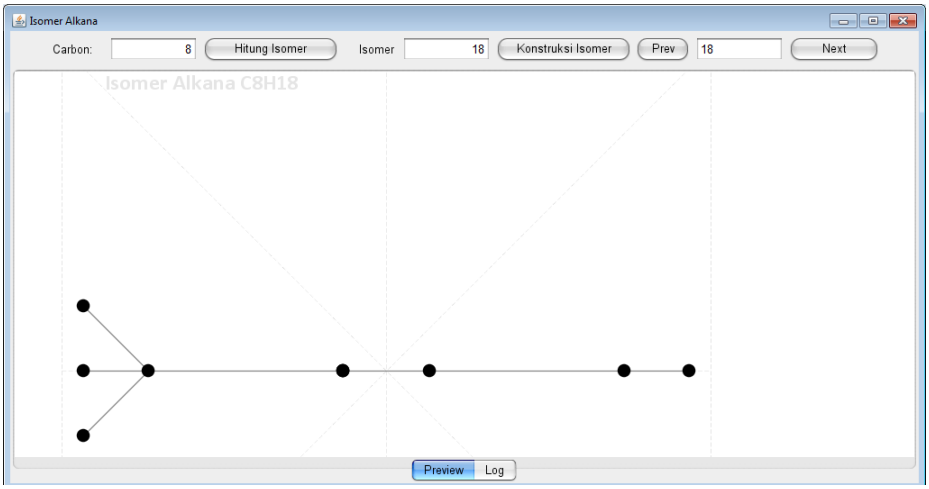
Gambar : Isomer-15 C8H18



Gambar : Isomer-16 C8H18



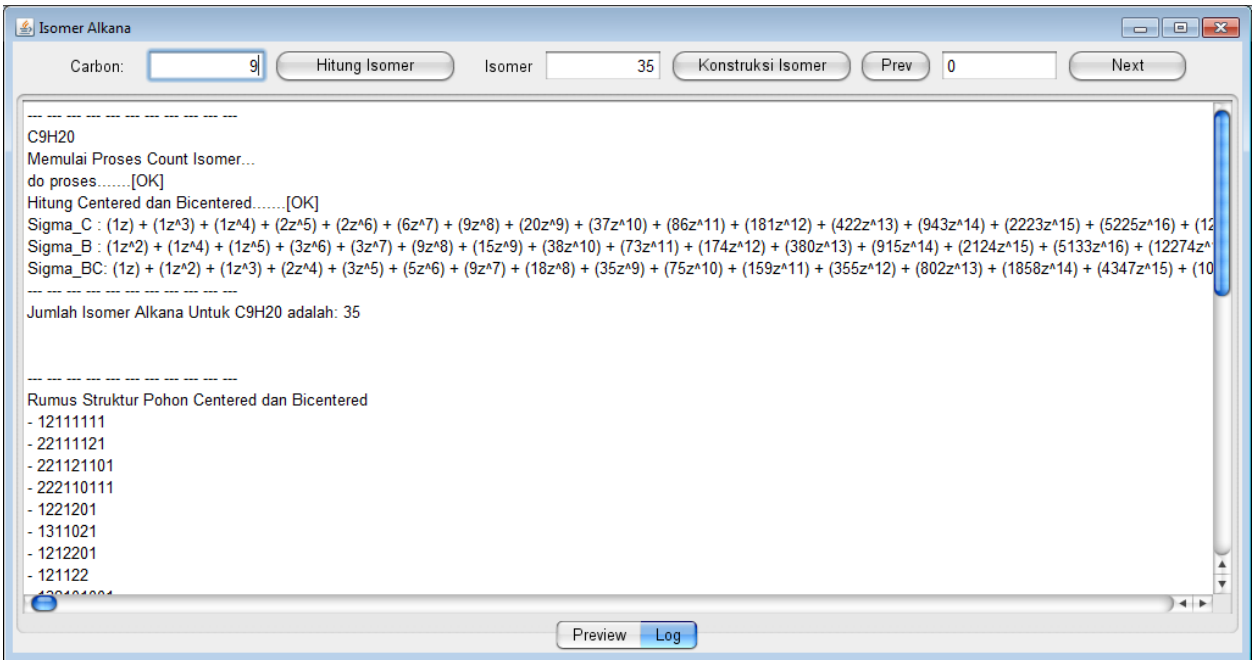
Gambar : Isomer-17 C8H18



Gambar : Isomer-18 C8H18

4. Pengujian keempat

Input: Carbon = 9



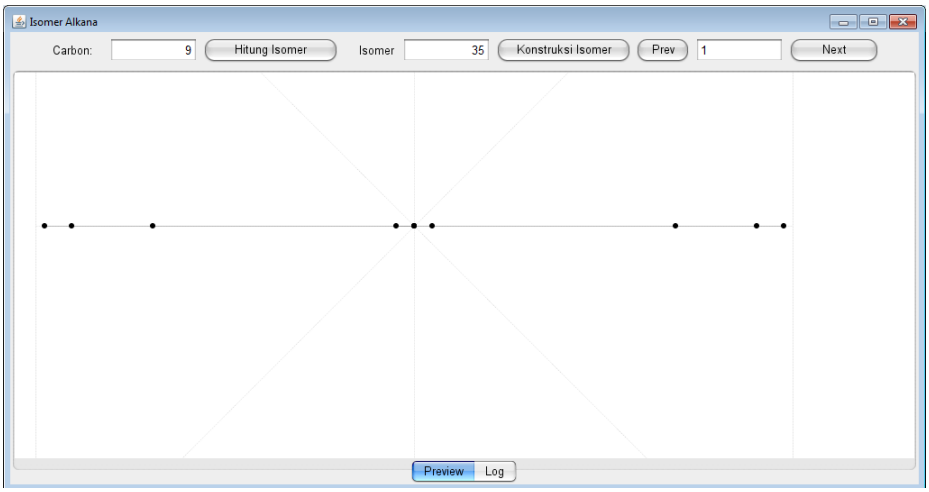
Diperoleh output: Jumlah Isomer Alkana Untuk C9H20 adalah: 35

Rumus Struktur Pohon Centered dan Bicentered

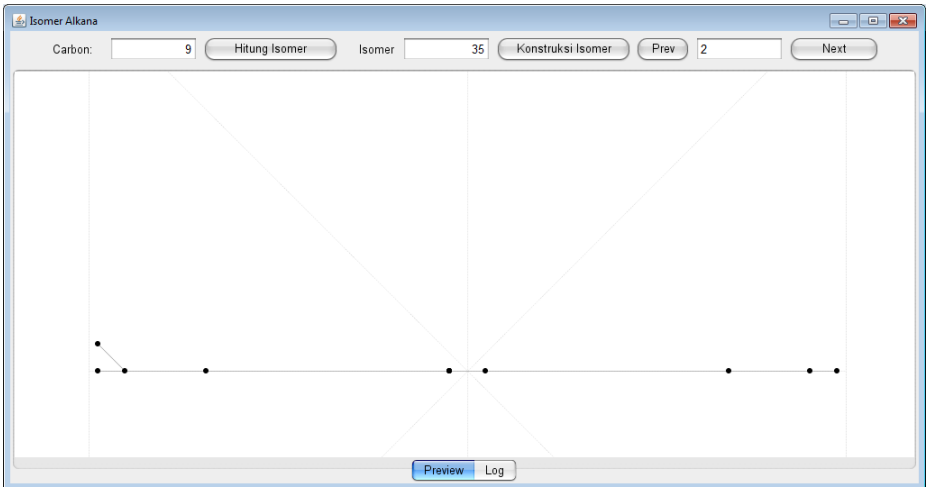
- 12111111
- 22111121
- 221121101
- 222110111
- 1221201
- 1311021
- 1212201
- 121122
- 132101001
- 12221001
- 121131
- 12311001
- 14110011
- 1221111
- 13111110

- 22222010
- 22212011
- 2221301
- 2212310
- 221132
- 22312001
- 22311011
- 223210010
- 22311002
- 2221211
- 22221110
- 2221112
- 143100
- 123300
- 13320
- 142200
- 142101
- 13221
- 13311
- 141111

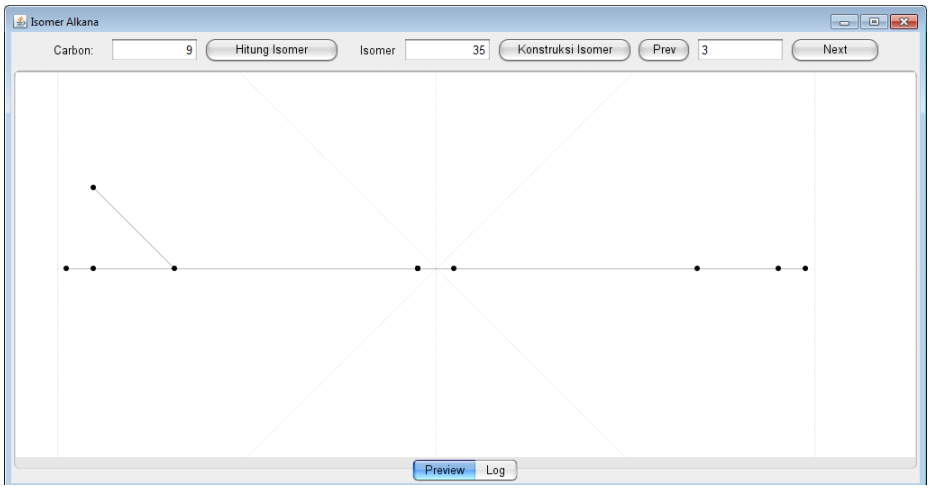
Preview Isomer:



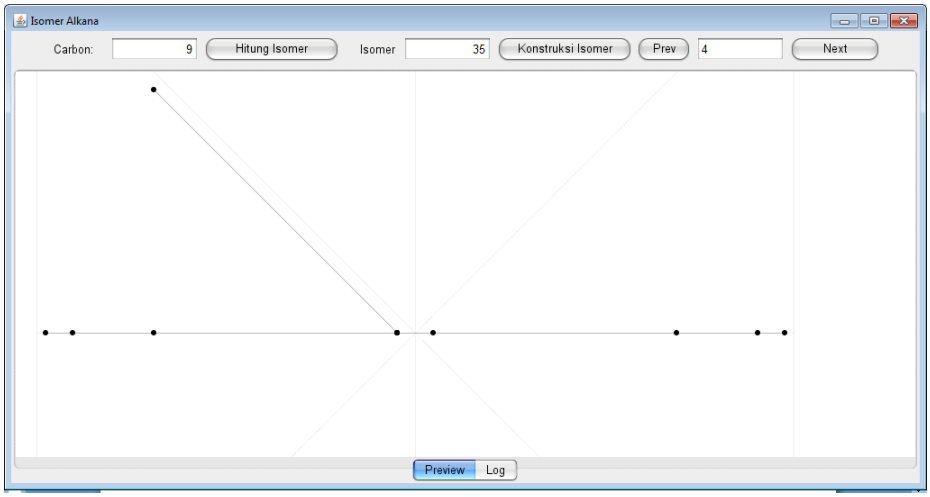
Gambar : Isomer-1 C9H20



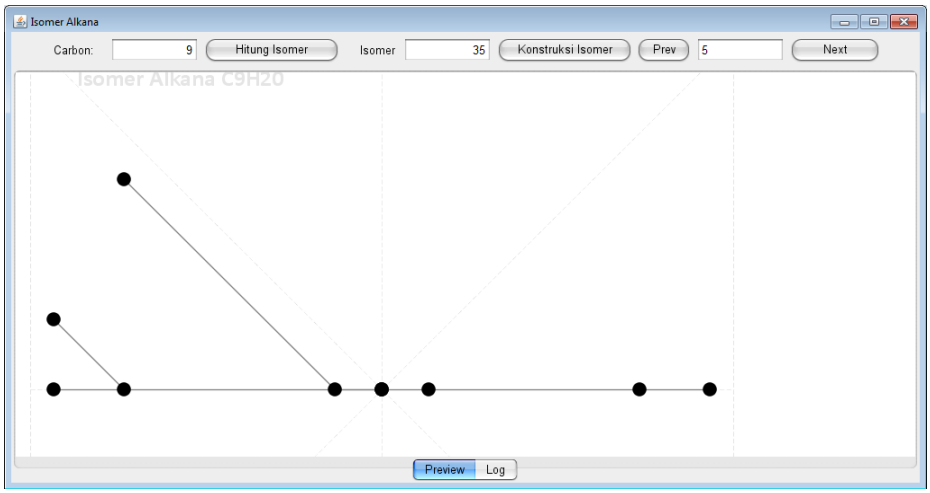
Gambar : Isomer-2 C9H20



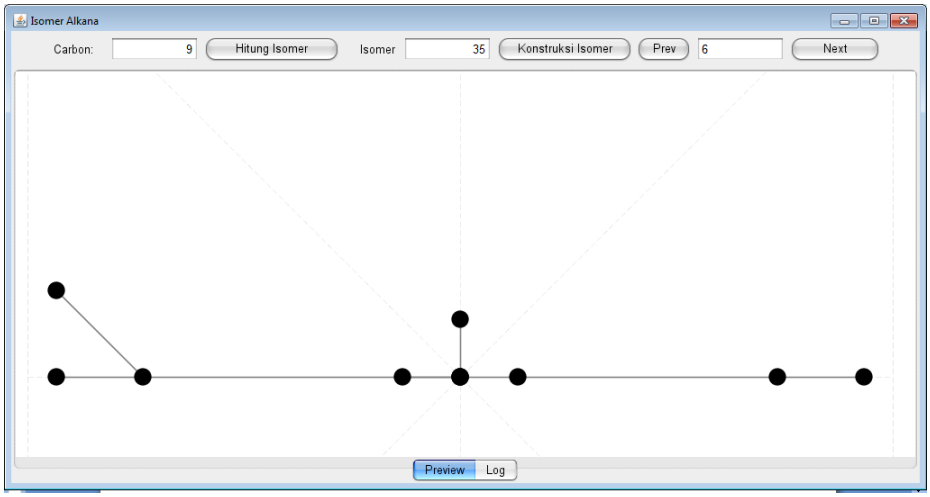
Gambar : Isomer-3 C9H20



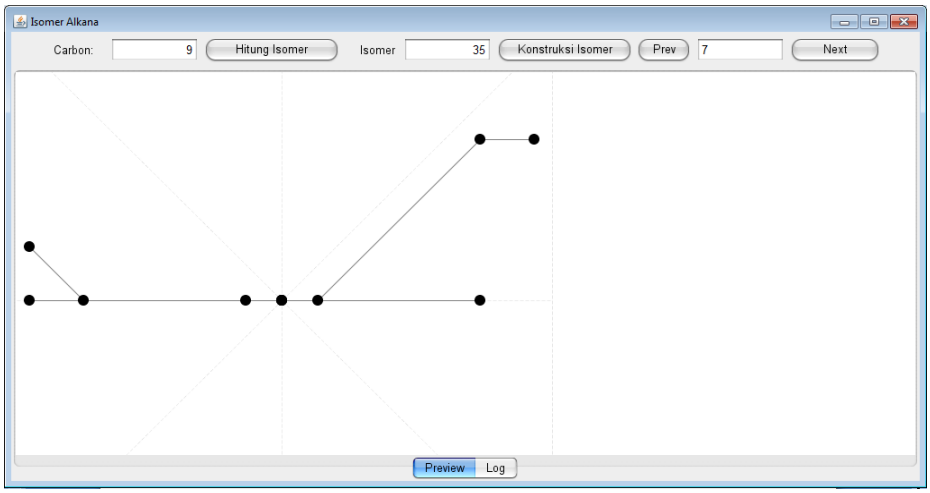
Gambar : Isomer-4 C9H20



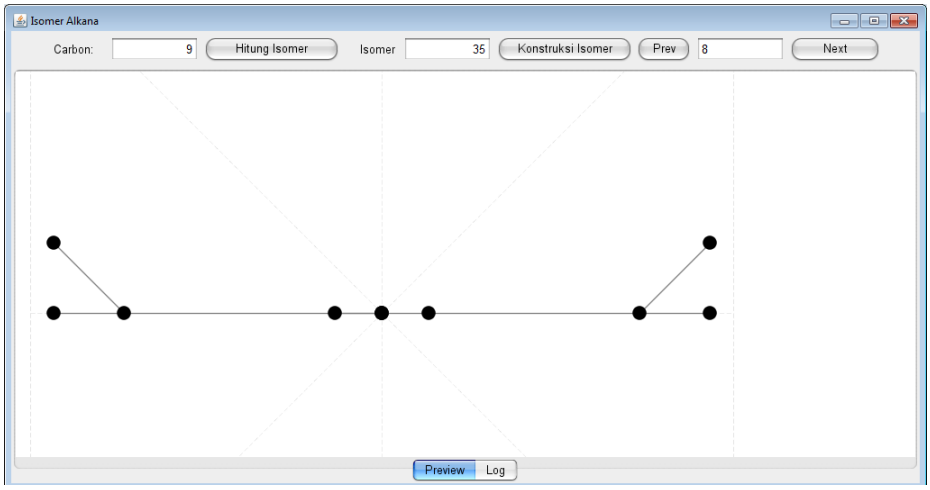
Gambar : Isomer-5 C9H20



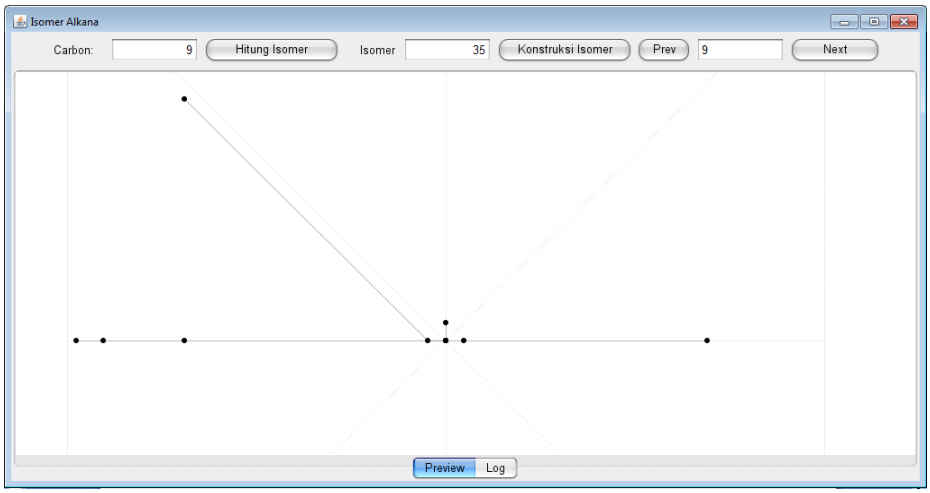
Gambar : Isomer-6 C9H20



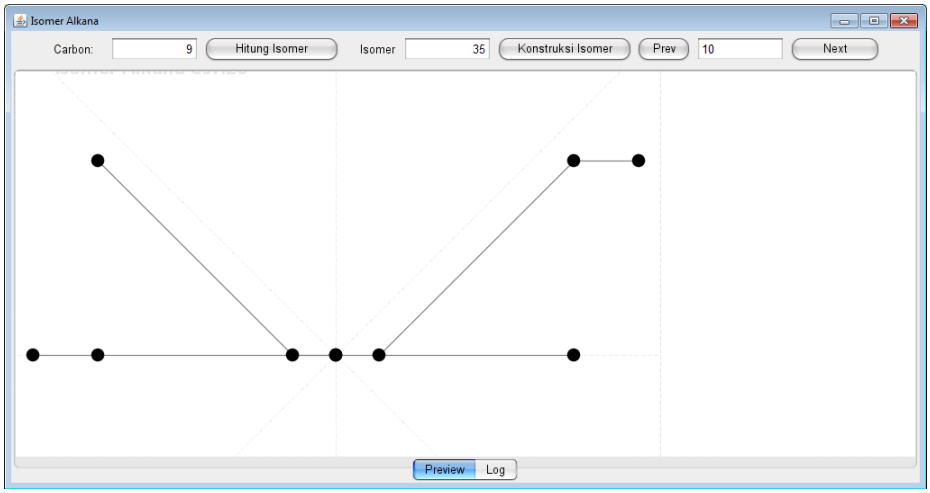
Gambar : Isomer-7 C9H20



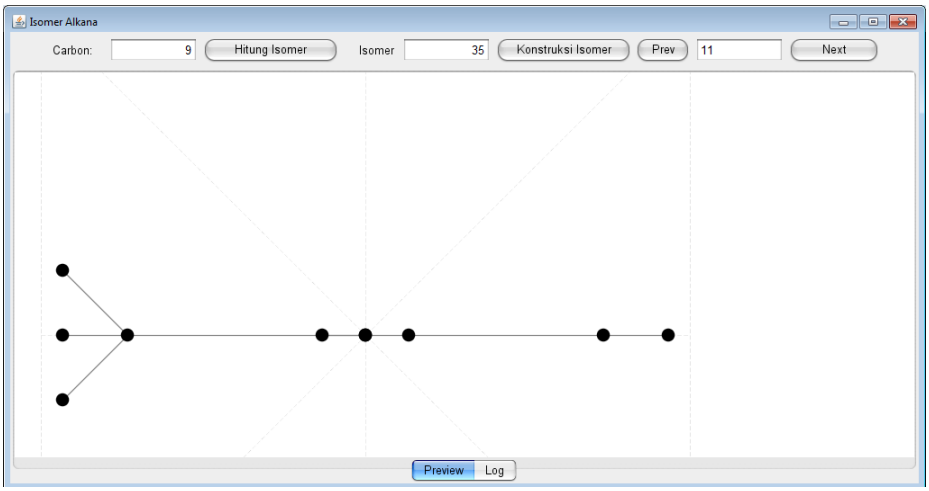
Gambar : Isomer-8 C9H20



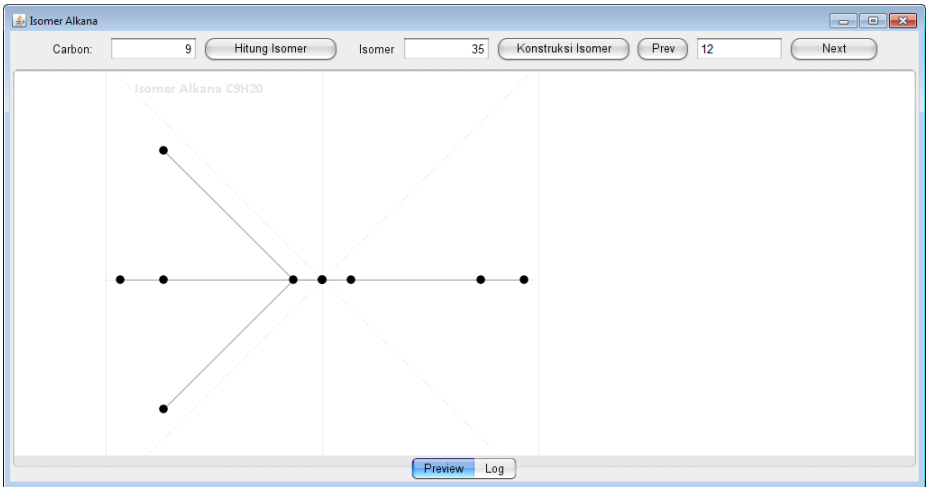
Gambar : Isomer-9 C9H20



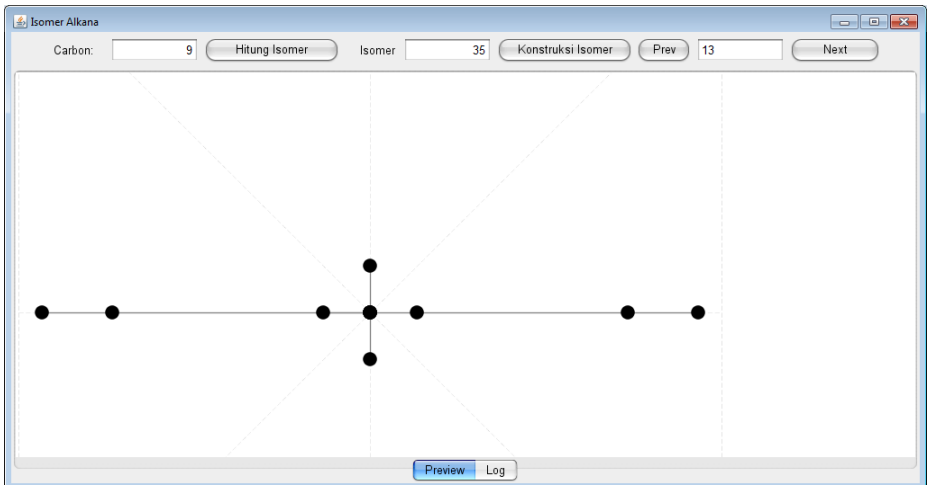
Gambar : Isomer-10 C9H20



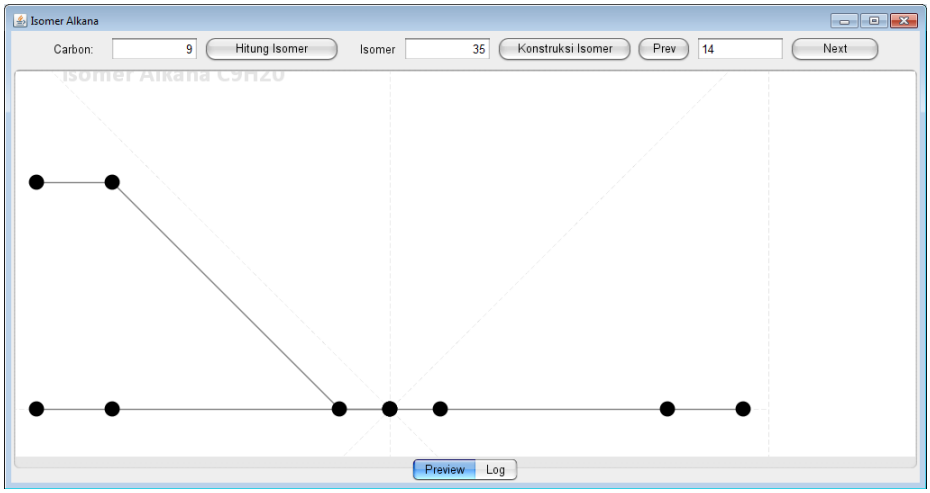
Gambar : Isomer-11 C9H20



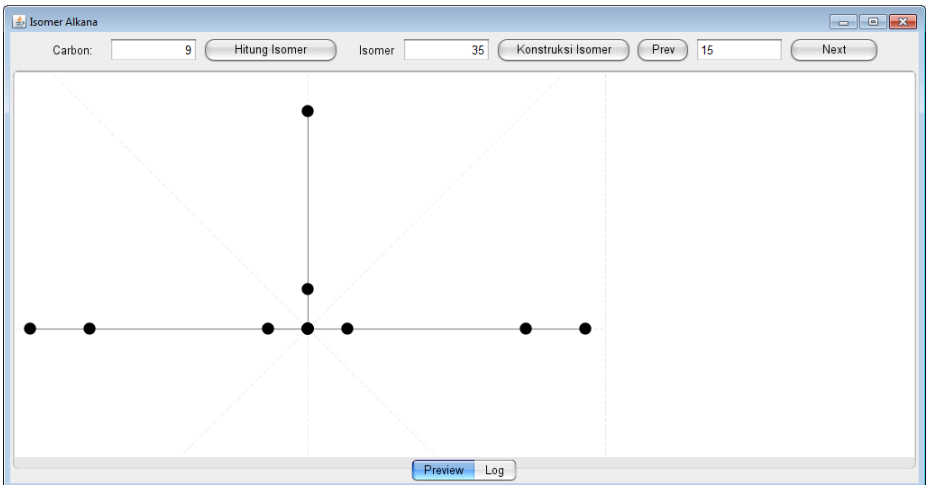
Gambar : Isomer-12 C9H20



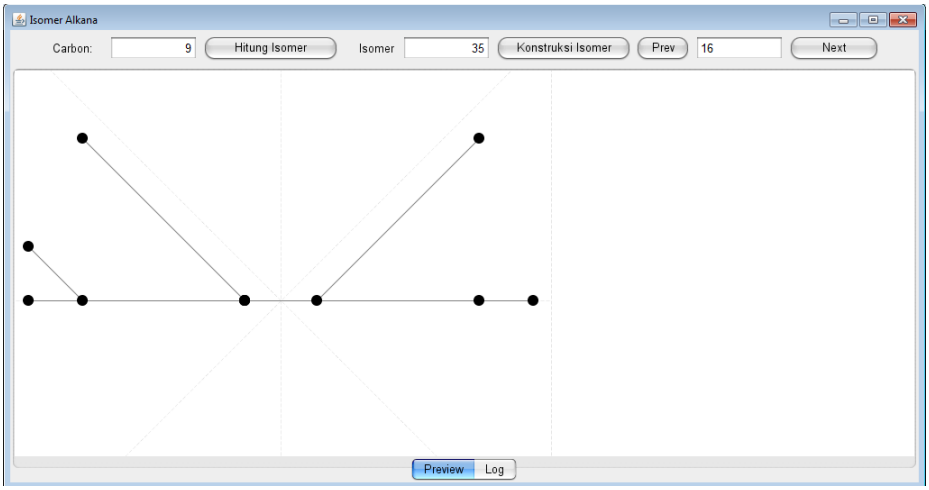
Gambar : Isomer-13 C9H20



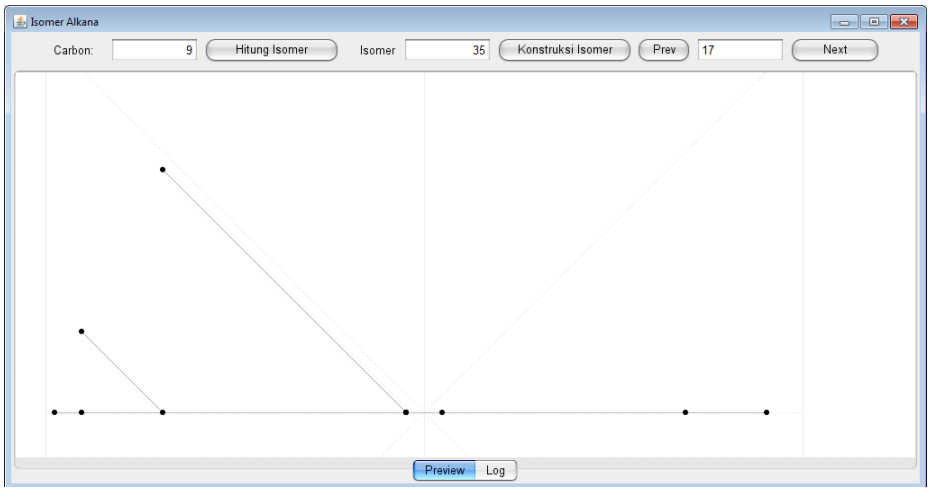
Gambar : Isomer-14 C9H20



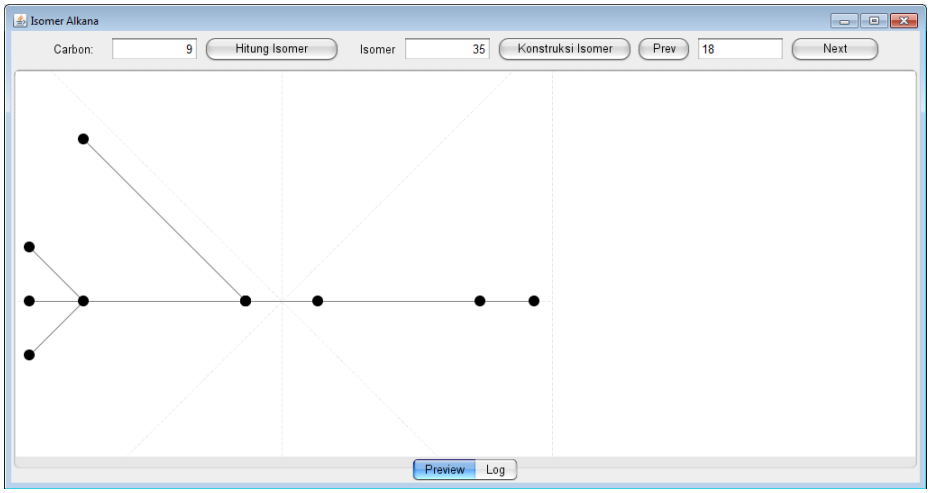
Gambar : Isomer-15 C9H20



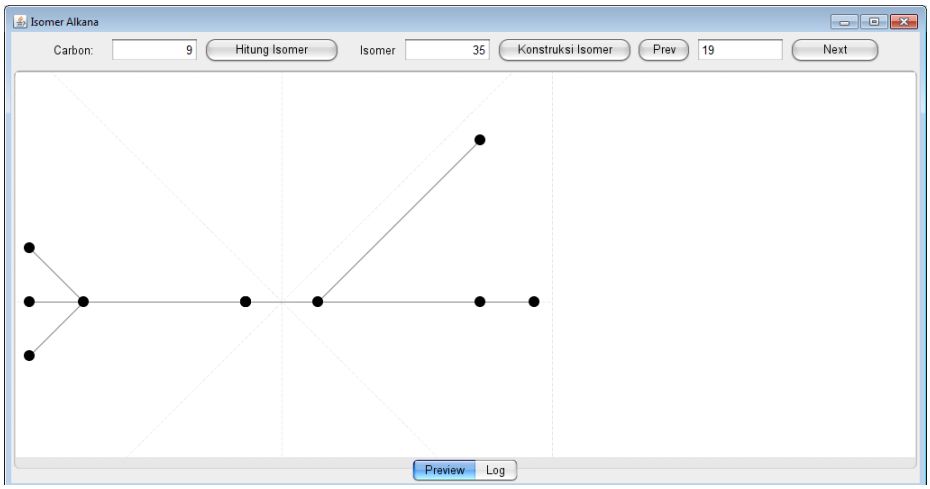
Gambar : Isomer-16 C9H20



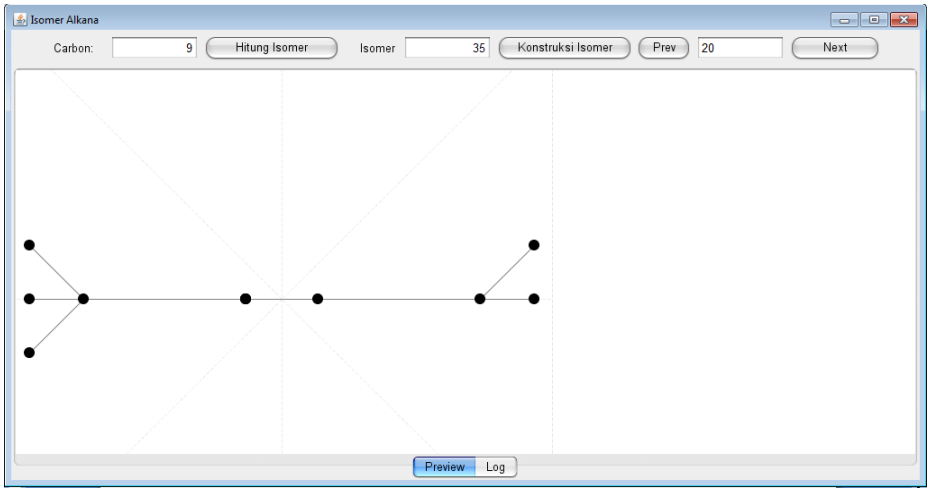
Gambar : Isomer-17 C9H20



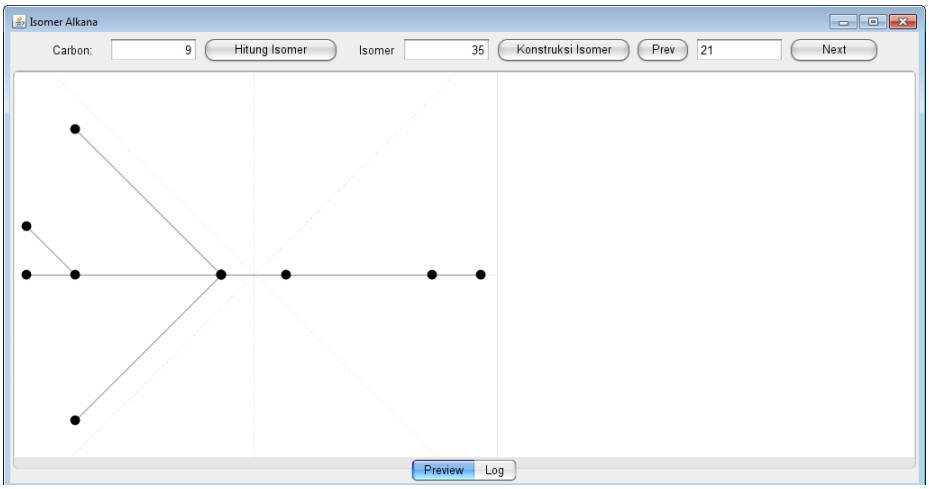
Gambar : Isomer-18 C9H20



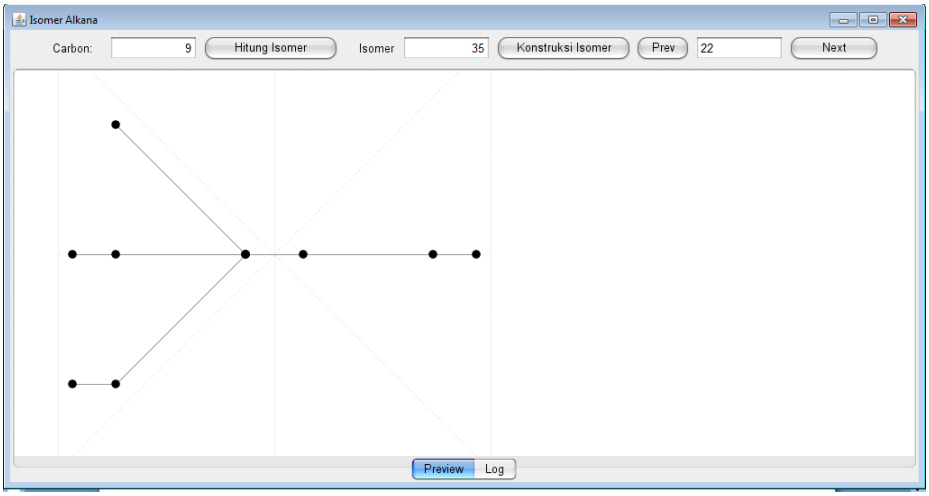
Gambar : Isomer-19 C9H20



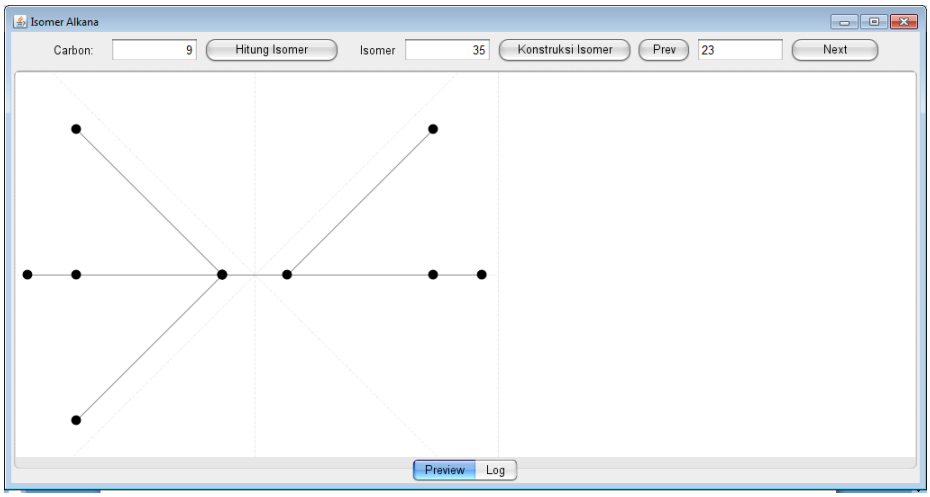
Gambar : Isomer-20 C9H20



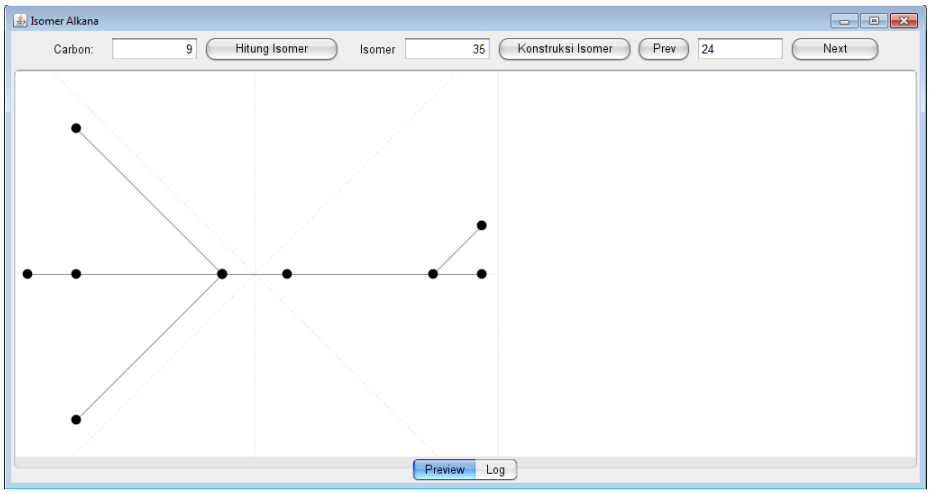
Gambar : Isomer-21 C9H20



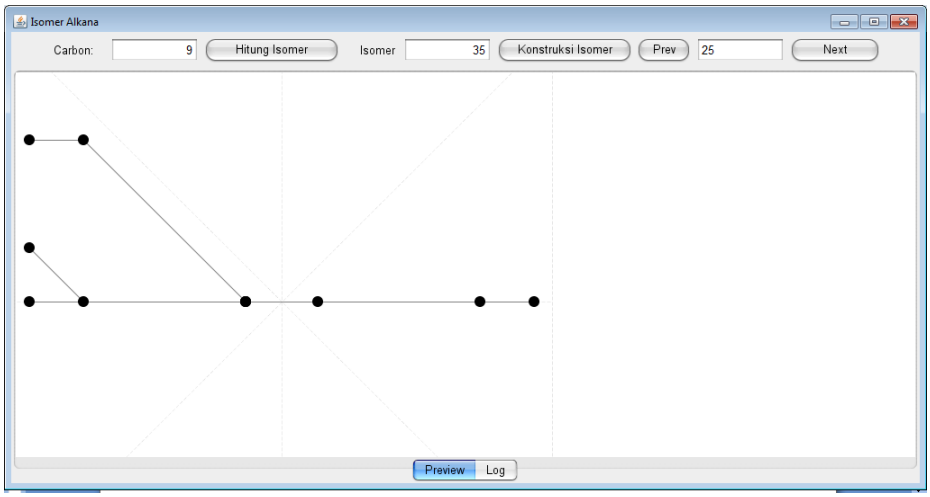
Gambar : Isomer-22 C9H20



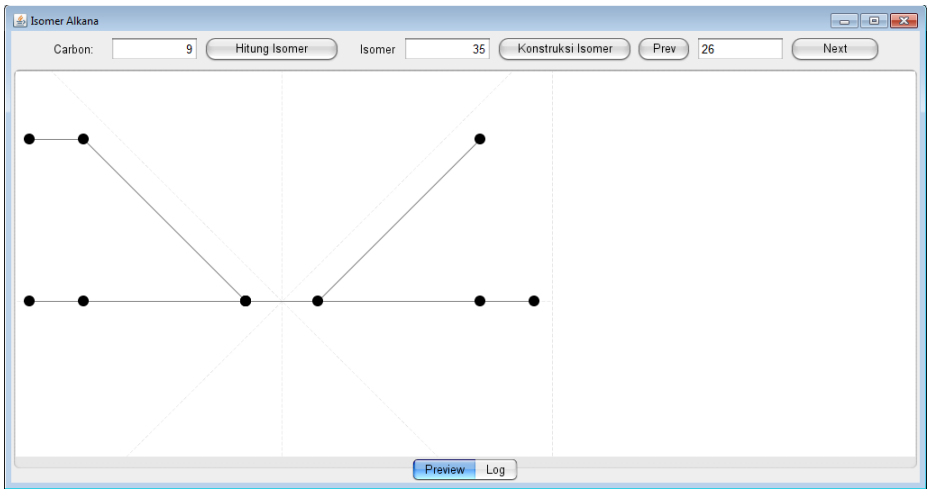
Gambar : Isomer-23 C9H20



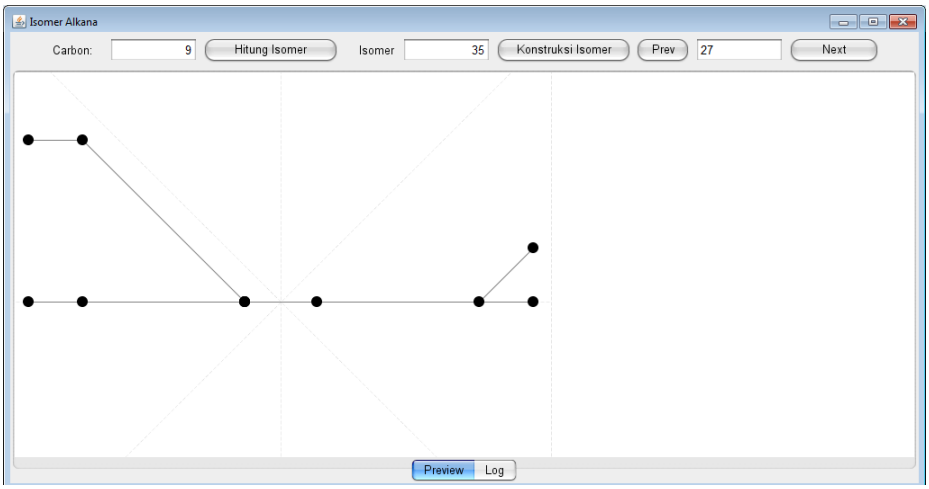
Gambar : Isomer-24 C9H20



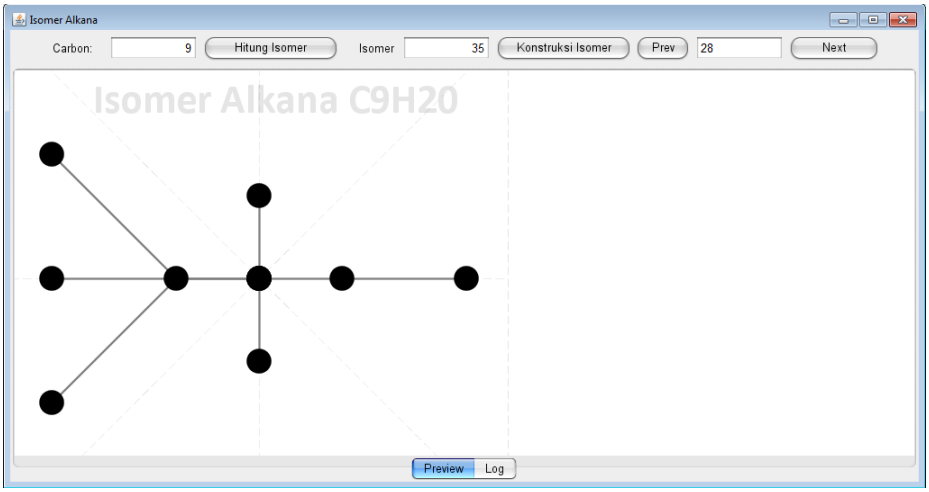
Gambar : Isomer-25 C9H20



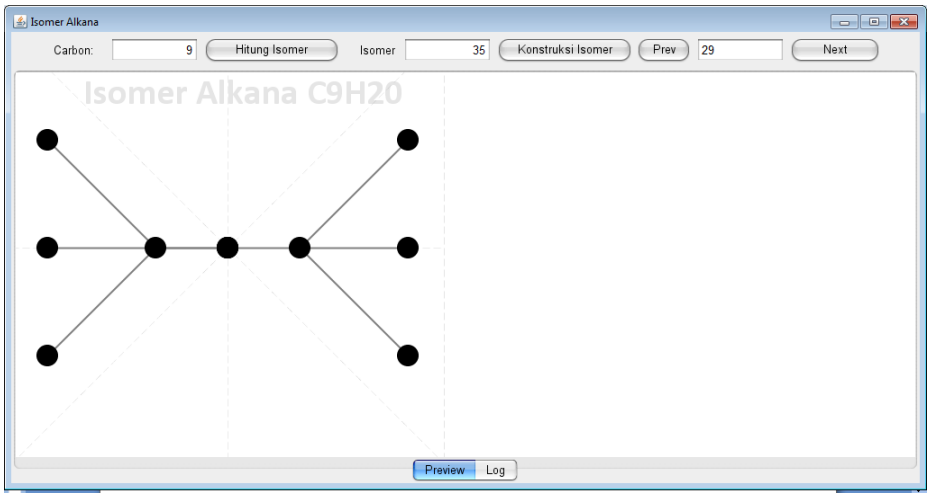
Gambar : Isomer-26 C9H20



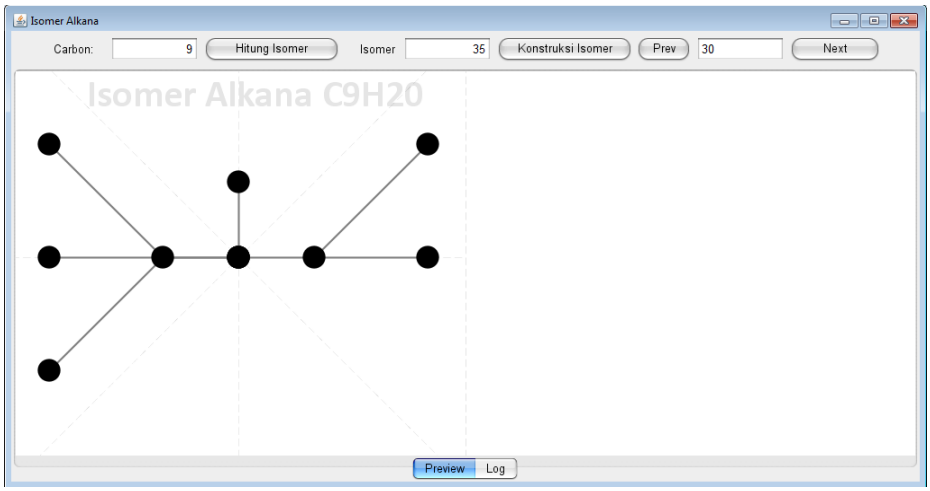
Gambar : Isomer-27 C9H20



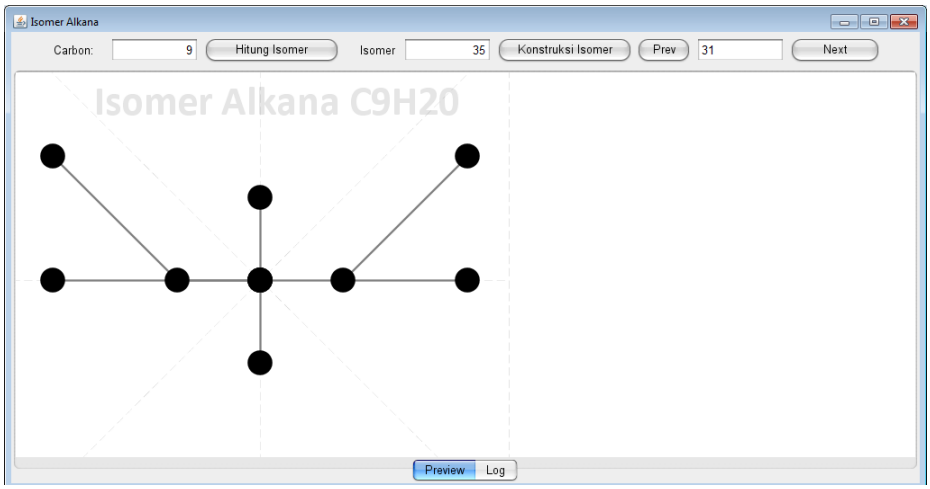
Gambar : Isomer-28 C9H20



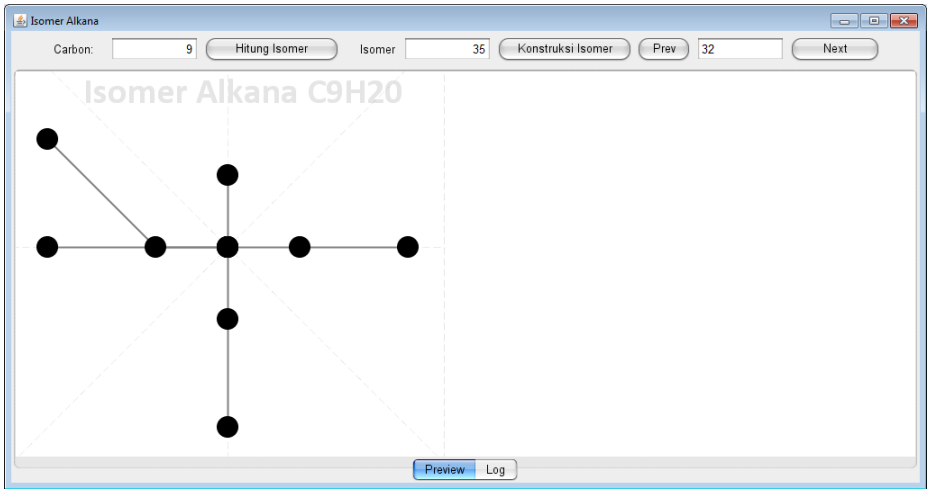
Gambar : Isomer-29 C9H20



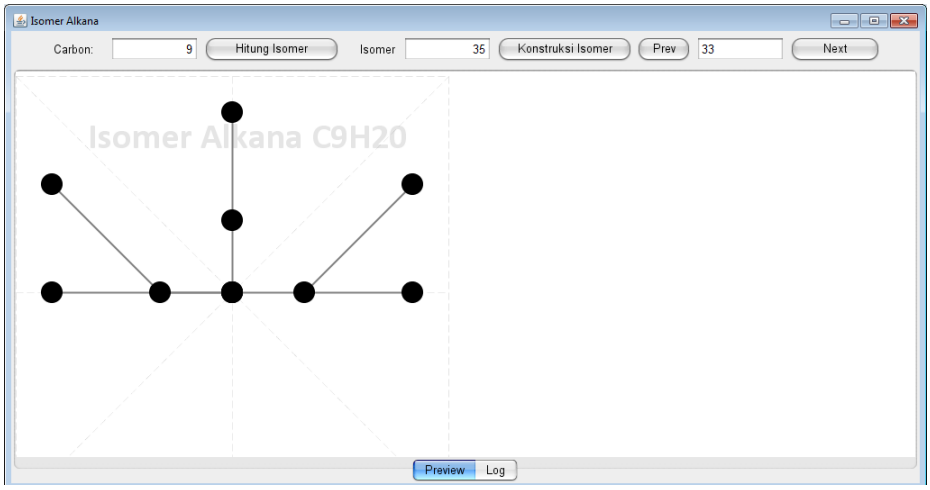
Gambar : Isomer-30 C9H20



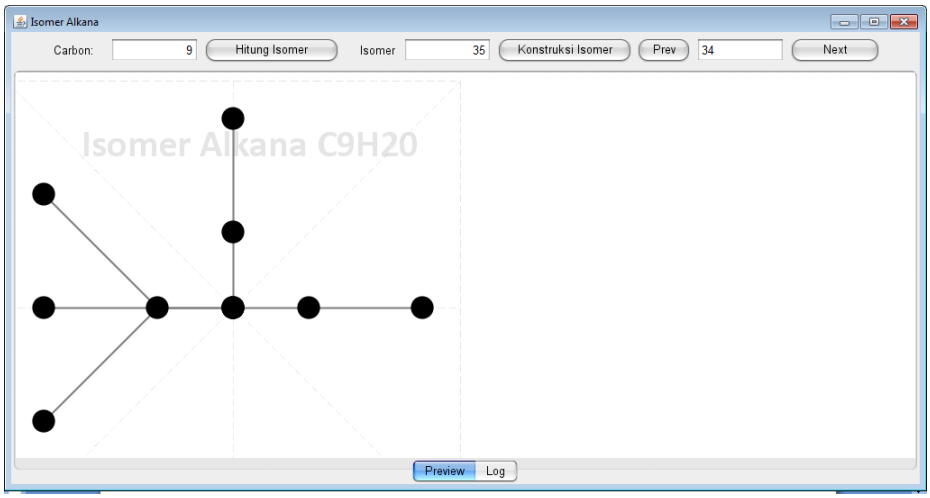
Gambar : Isomer-31 C9H20



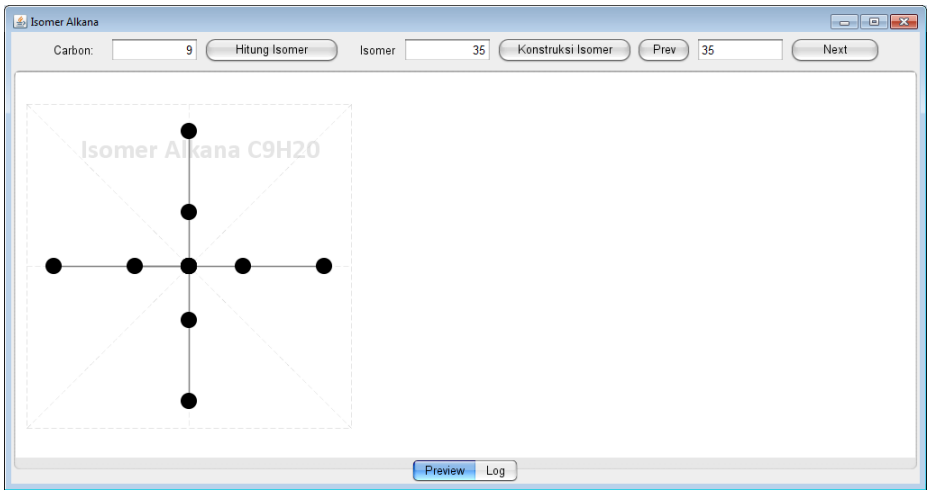
Gambar : Isomer-32 C9H20



Gambar : Isomer-33 C9H20



Gambar : Isomer-34 C9H20



Gambar : Isomer-35 C9H20

1. InterfacePolynomialZet

```
package com.alkana.Isomer


public interface InterfacePolynomialZet {

    public void insert(int c, int p);//c z^p
    public void insert(int[]z);//c z^p
    public int[]getZet();
    public void empty();
    public void plus(int[]z);
    public void minus(int[]z);
    public void times(int[]z);
    public void power(int p);//(f(z))^p
    public void powerZet(int p);//f((z)^p)
```

2. PolynomialZet.java

```
package com.alkana.isomer

public class PolynomialZet implements InterfacePolynomialZet{

    private int MAX                 = 1000000;
    private int[]coef       = new int[MAX];
    private int orde        = -1;

    @Override
    public void insert(int c, int p) {
            if((p>-1)&&(p<MAX)){
                    if(p>this.orde){this.orde      = p;}
                    this.coef[p]= this.coef[p]+c;
            }
    }//end of insert
    @Override
    public void insert(int[] z) {
            if(z!=null){
                    for(int i=0;i<z.length;i++){
                            this.insert(z[i],i);
    @Override
    public int[] getZet() {
            if(this.orde<0){return null;}
            else{
                    int size        = 1+this.orde;
                    int[]result     = new int[size];
                    for(int i=0;i<result.length;i++){
                            result[i]       = this.coef[i];
```

```
                return result;

    public int getValueAt(int index){
            int value       = 0;
            if((this.orde<0)||(index<0)||(index>this.MAX)){value     = 0;}
            else{
                    value  = this.coef[index];
    return value;
    }//end of getValueAt(int index)

    @Override
    public String toString(){
            String result  ="";
            int[]poly       = this.getZet();
            if(poly==null){
                    result  = "null";
            }else{
                    for(int i=0;i<poly.length;i++){
                            if(poly[i]!=0){
                                    if(i==0){ result                                =
result.concat("("+(poly[i])+")");}
                                    else if(i==1){result                    =
result.concat("("+(poly[i])+"z)");}
                                    else{result                                             =
result.concat("("+(poly[i])+"z^"+(i)+")");}
                                    if(i<(-1+poly.length)){result = result.concat(" + ");}
result      = result.concat(" --> orde: "+(this.orde));
return result;

@Override
public void empty() {
```

```
        this.coef       = new int[MAX];
        this.orde       = -1;
@Override
public void plus(int[] z) {
        if(z!=null){
                for(int i=0;i<z.length;i++){
                        this.insert(z[i], i);
@Override
public void minus(int[] z) {
        if(z!=null){
                for(int i=0;i<z.length;i++){
                        this.insert((-1*z[i]), i);
                for(int i=this.orde;i>0;i--){
                        if(this.coef[i]!=0){
                                this.orde       = i;
                                break;
@Override
public void times(int[] z) {
        int[]a  = this.getZet();
        int[]b  = z.clone();
        if((a!=null)&&(b!=null)){
                int size        = 1+(-1+a.length)+(-1+b.length);
                int[]result     = new int[size];
                for(int i=0;i<a.length;i++){
                        for(int j=0;j<b.length;j++){
                                int index               = i+j;
                                int value               = a[i]*b[j];
                                result[index]  = result[index]+value;
                this.empty();
                for(int i=0;i<result.length;i++){
                        this.insert(result[i], i);
```

```
@Override
    public void power(int p) {
            int[]a=this.getZet();
            if(a!=null){
                    if(p>1){
                            int[]b   = a.clone();
                            int size= -1+p;
                            for(int i=0;i<size;i++){
                                    this.times(b);
    @Override
    public void powerZet(int p) {
            if(p>=0){
                    int[]a   = this.getZet();
                    if(a!=null){
                            int size          = 1+(-1+a.length)*p;
                            int[]result       = new int[size];
                            for(int i=0;i<a.length;i++){
                                    int index        = p*i;
                                    result[index]  = result[index]+a[i];
                            this.empty();
                            for(int i=0;i<result.length;i++){
                                    this.insert(result[i], i);
    /**
     * @param args
     */
    public static void main(String[] args) {
            PolynomialZet pz     = new PolynomialZet();
            pz.insert(12, 2);
            pz.insert(14, 4);
            pz.insert(10, 0);
            //pz.insert(-20, 2);
```

```
System.out.println(pz.toString());
int[]z1 = {0,11,0,13,0,15};
pz.plus(z1);
System.out.println(pz.toString());
int[]z2 = {0,11,0,13,0,15};
pz.minus(z2);
System.out.println(pz.toString());
int[]z3 = {2};
pz.times(z3);
System.out.println(pz.toString());
int[]z4 = {0,1};
pz.times(z4);
System.out.println(pz.toString());
int[]z5 = {1,1,1};
pz.times(z5);
System.out.println(pz.toString());
pz.powerZet(0);
System.out.println(pz.toString());
int[]z6 = {210};
pz.minus(z6);
pz.insert(3, 1);
System.out.println(pz.toString());
pz.power(3);
System.out.println(pz.toString());
```

3. S2.java

```
package com.akana.isomer;


public class S2 {

    private int[]Tz                                 = null;
    public static int[]result        = null;

    public S2(int[] tz) {
            super();
            Tz = tz;
            this.evaluateS2();
    }

    private void evaluateS2(){
            if(this.Tz!=null){
                    PolynomialZet p1     = new PolynomialZet();
                    p1.insert(this.Tz.clone());
                    p1.power(2);

                    PolynomialZet p2     = new PolynomialZet();
                    p2.insert(this.Tz.clone());
                    p2.powerZet(2);

                    p1.plus(p2.getZet());

                    int[]r    = p1.getZet().clone();

                    for(int i=0;i<r.length;i++){
```

```
                        r[i]=r[i]/2;
                }

                this.result=r.clone();

        }else{
                this.result             = null;
        }
    }

    public String toString(){
        String sresult ="";
        if(this.result==null){
                sresult= "null";
        }else{
                for(int i=0;i<this.result.length;i++){
                        if(this.result[i]!=0){
                                if(i==0){ sresult
    = sresult.concat("("+(this.result[i])+")");}
                                else if(i==1){sresult                                   =
sresult.concat("("+(this.result[i])+"z)");}
                                else{sresult
    = sresult.concat("("+(this.result[i])+"z^"+(i)+")");}
                                if(i<(-1+this.result.length)){sresult =  sresult.concat("  +
");}
                        }
                }
                sresult= sresult.concat(" --> orde: "+(-1+this.result.length));
        }
        return sresult;
    }
```

```
public static void main(String[] args) {
        PolynomialZet pz     = new PolynomialZet();
        //pz.insert(1, 0);
        //pz.insert(1, 1);
        pz.insert(1, 2);
        pz.insert(1, 3);
        pz.insert(1, 4);
        System.out.println(pz.toString());
        S2 s2  = new S2(pz.getZet());
        System.out.println(s2.toString());
```

4. S3.java

```
package com.alakana.isomer;

public class S3 {

    private int[]Tz                                 = null;
    public static int[]result        = null;



    public S3(int[] tz) {
            super();
            Tz = tz;
            this.evaluateS3();
    }

    private void evaluateS3(){
            if(this.Tz!=null){
                    PolynomialZet p1     = new PolynomialZet();
                    p1.insert(this.Tz.clone());
                    p1.power(3);

                    PolynomialZet p2     = new PolynomialZet();
                    p2.insert(this.Tz.clone());
                    int[]z0            = {3};
                    p2.times(z0);

                    PolynomialZet p3     = new PolynomialZet();
                    p3.insert(this.Tz.clone());
                    p3.powerZet(2);
                    p2.times(p3.getZet());
```

```
                PolynomialZet p4    = new PolynomialZet();
                p4.insert(this.Tz.clone());
                p4.powerZet(3);
                int[]z1         = {2};
                p4.times(z1);
                p1.plus(p2.getZet());
                p1.plus(p4.getZet());
                int[] pembilang     = p1.getZet();
                int[] r             = new int[pembilang.length];
                for(int i=0;i<pembilang.length;i++){r[i]=pembilang[i]/6;}
                this.result         = r.clone();
        }else{
                this.result         = null;
        }
    }

    public String toString(){
        String sresult ="";
        if(this.result==null){
                sresult= "null";
        }else{
                for(int i=0;i<this.result.length;i++){
                        if(this.result[i]!=0){
                                if(i==0){ sresult
    = sresult.concat("("+(this.result[i])+")");}
                                else if(i==1){sresult                                =
sresult.concat("("+(this.result[i])+"z)");}
                                else{sresult
    = sresult.concat("("+(this.result[i])+"z^"+(i)+")");}
```

```
                                    if(i<(-1+this.result.length)){sresult =  sresult.concat("  +
");;}
                                }
                        }
                        sresult= sresult.concat(" --> orde: "+(-1+this.result.length));
                }
                return sresult;
        }


        /**
         * @param args
         */
        public static void main(String[] args) {
                PolynomialZet pz     = new PolynomialZet();
                pz.insert(1, 0);
                pz.insert(1, 1);
                pz.insert(1, 2);
                pz.insert(1, 3);
                pz.insert(1, 4);
                System.out.println(pz.toString());
                S3 s3  = new S3(pz.getZet());
                System.out.println(s3.toString());
```

5. S4.java

```
package com.alakana.isomer;

public class S4 {

    private int[]Tz                                 = null;
    public static int[]result        = null;



    public S4(int[] tz) {
            super();
            Tz = tz;
            this.evaluateS4();
    }

    private void evaluateS4(){
            if(this.Tz!=null){
                    PolynomialZet p1     = new PolynomialZet();
                    p1.insert(this.Tz.clone());
                    p1.power(4);

                    PolynomialZet p2     = new PolynomialZet();
                    p2.insert(this.Tz.clone());
                    p2.powerZet(2);
                    int[]z0={6};
                    p2.times(z0);

                    PolynomialZet p3     = new PolynomialZet();
                    p3.insert(this.Tz.clone());
                    p3.power(2);
```

```
p2.times(p3.getZet());

PolynomialZet p4     = new PolynomialZet();
p4.insert(this.Tz.clone());
p4.powerZet(3);
int[]z1={8};
p4.times(z1);

PolynomialZet p5     = new PolynomialZet();
p5.insert(this.Tz.clone());
p4.times(p5.getZet());

PolynomialZet p6     = new PolynomialZet();
p6.insert(this.Tz.clone());
p6.powerZet(2);
p6.power(2);
int[]z2={3};
p6.times(z2);

PolynomialZet p7     = new PolynomialZet();
p7.insert(this.Tz.clone());
p7.powerZet(4);
int[]z3={6};
p7.times(z3);

p1.plus(p2.getZet());
p1.plus(p4.getZet());
p1.plus(p6.getZet());
p1.plus(p7.getZet());

int[] r=p1.getZet().clone();
```

```
                for(int i=0;i<r.length;i++){
                        r[i]=r[i]/24;
                }
                this.result=r.clone();

        }else{
                this.result             = null;
        }
    }

    public String toString(){
        String sresult ="";
        if(this.result==null){
                sresult= "null";
        }else{
                for(int i=0;i<this.result.length;i++){
                        if(this.result[i]!=0){
                                if(i==0){ sresult
    = sresult.concat("("+(this.result[i])+")");}
                                else if(i==1){sresult                                   =
sresult.concat("("+(this.result[i])+"z)");}
                                else{sresult
    = sresult.concat("("+(this.result[i])+"z^"+(i)+")");}
                                if(i<(-1+this.result.length)){sresult =  sresult.concat("  +
");}
                        }
                }
                sresult= sresult.concat(" --> orde: "+(-1+this.result.length));
        }
        return sresult;
```

```
}

/**
 * @param args
 */
public static void main(String[] args) {
        PolynomialZet pz     = new PolynomialZet();
        pz.insert(1, 0);
        pz.insert(1, 1);
        pz.insert(1, 2);
        pz.insert(1, 3);
        pz.insert(1, 4);
        System.out.println(pz.toString());
        S4 s4  = new S4(pz.getZet());
        System.out.println(s4.toString());
```

6. Carbon.java

```
package com.alakana.isomer;

public class Carbon {
    String address;
    int x;
    int y;
    int MAX_RADIUS;
    int type;
    int kuadran;
    int radius;
    int numAnak;
    final int MAX_NUM_ANAK = 3;
    boolean visited        = false;

    Carbon c0     = null;

    Carbon c1     = null;
    Carbon c2     = null;
    Carbon c3     = null;

    int space;
    int fixedPoint;
    int newPoint;

    public Carbon(String address, int x, int y, int MAX_RADIUS) {
            super();
            this.address            = address;
            this.x                          = x;
            this.y                          = y;
```

```
        this.MAX_RADIUS  = MAX_RADIUS;

        this.type                   =
Integer.parseInt(String.valueOf(address.charAt(0)));
        this.kuadran        = Integer.parseInt(String.valueOf(address.charAt(1)));
        this.radius                 = address.length()-2;
        this.numAnak                = 0;
        this.space                  =   (int)   Math.pow(3,   (this.MAX_RADIUS-
this.radius-1));
        //System.out.println("space :"+(space));
        if(this.kuadran==1||this.kuadran==2){
            this.fixedPoint     = this.y;
        }else if(this.kuadran==3||this.kuadran==4){
            this.fixedPoint     = this.x;
        }//end      of      if(this.kuadran==1||this.kuadran==2)      and      else
if(this.kuadran==3||this.kuadran==4)

        if(this.kuadran==1){
            this.newPoint= this.x-this.space;
        }else if(this.kuadran==2){
            this.newPoint= this.x+this.space;
        }else if(this.kuadran==3){
            this.newPoint= this.y-this.space;
        }else if(this.kuadran==4){
            this.newPoint= this.y+this.space;
        }
    }//end of constructor

    public void saveParent(){
        if(this.c1!=null){this.c1.c0   = this;}
        if(this.c2!=null){this.c2.c0   = this;}
```

```
        if(this.c3!=null){this.c3.c0    = this;}
    }//end of saveParent()

    public void addChild(int num){
        if((num>0)&&(num<4)){
            for(int i=0;i<num;i++){
                this.addChild();
            }
        }
    }//end of addChild

    public void addChild(){
        String address0      = this.address;
        int newX,newY;
        if(this.radius>=0&&this.radius<=this.MAX_RADIUS){
            if(this.kuadran==1){
                newX  = this.newPoint;
                if(this.numAnak==0){
                    address0      = address0.concat("1");
                    newY          = this.fixedPoint;
                    this.c1=    new    Carbon(address0,    newX,    newY,
MAX_RADIUS);
                    this.numAnak++;
                }else if(this.numAnak==1){
                    address0      = address0.concat("2");
                    newY          = this.fixedPoint-this.space;
                    this.c2=    new    Carbon(address0,    newX,    newY,
MAX_RADIUS);
                    this.numAnak++;
                }else if(this.numAnak==2){
                    address0      = address0.concat("3");
```

```
                    newY        = this.fixedPoint+this.space;
                    this.c3=    new    Carbon(address0,    newX,    newY,
MAX_RADIUS);
                    this.numAnak++;
                }//end of if(this.numAnak==0) else if
            }else if(this.kuadran==2){
                newX  = this.newPoint;
                if(this.numAnak==0){
                    address0    = address0.concat("1");
                    newY        = this.fixedPoint;
                    this.c1=    new    Carbon(address0,    newX,    newY,
MAX_RADIUS);
                    this.numAnak++;
                }else if(this.numAnak==1){
                    address0    = address0.concat("2");
                    newY        = this.fixedPoint-this.space;
                    this.c2=    new    Carbon(address0,    newX,    newY,
MAX_RADIUS);
                    this.numAnak++;
                }else if(this.numAnak==2){
                    address0    = address0.concat("3");
                    newY        = this.fixedPoint+this.space;
                    this.c3=    new    Carbon(address0,    newX,    newY,
MAX_RADIUS);
                    this.numAnak++;
                }//end of if(this.numAnak==0) else if
            }else if(this.kuadran==3){
                newY  = this.newPoint;
                if(this.numAnak==0){
                    address0    = address0.concat("1");
                    newX        = this.fixedPoint;
```

```
                    this.c1= new Carbon(address0, newX, newY,
MAX_RADIUS);
                    this.numAnak++;
                }else if(this.numAnak==1){
                    address0    = address0.concat("2");
                    newX        = this.fixedPoint-this.space;
                    this.c2= new Carbon(address0, newX, newY,
MAX_RADIUS);
                    this.numAnak++;
                }else if(this.numAnak==2){
                    address0    = address0.concat("3");
                    newX        = this.fixedPoint+this.space;
                    this.c3= new Carbon(address0, newX, newY,
MAX_RADIUS);
                    this.numAnak++;
                }//end of if(this.numAnak==0) else if
            }else if(this.kuadran==4){
                newY  = this.newPoint;
                if(this.numAnak==0){
                    address0    = address0.concat("1");
                    newX        = this.fixedPoint;
                    this.c1= new Carbon(address0, newX, newY,
MAX_RADIUS);
                    this.numAnak++;
                }else if(this.numAnak==1){
                    address0    = address0.concat("2");
                    newX        = this.fixedPoint-this.space;
                    this.c2= new Carbon(address0, newX, newY,
MAX_RADIUS);
                    this.numAnak++;
                }else if(this.numAnak==2){
```

```
                                    address0       = address0.concat("3");
                                    newX           = this.fixedPoint+this.space;
                                    this.c3=    new    Carbon(address0,    newX,    newY,
MAX_RADIUS);
                                    this.numAnak++;
                                }//end of if(this.numAnak==0) else if
                            }//end of if(this.kuadran==1) else if
                        }//end of if(this.radius>=0&&this.radius<=this.MAX_RADIUS)
                        this.saveParent();
                    }//end of addChild()
                }//end f class Carbon
```

7. Canvas.java

```
package com.alakana.isomer;

import java.awt.AlphaComposite;
import java.awt.BasicStroke;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.Stroke;
import java.awt.geom.AffineTransform;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JPanel;


public class Canvas extends JPanel{

    double translateX;
    double translateY;
    double scale;

    int nC                    = 0;
    int gridSize    = 20;
    Carbon carbonX[]= null;

    Canvas() {
            this.translateX     = 0;
```

```
        this.translateY     = 0;
        this.scale          = 1;
        this.setOpaque(false);
        this.setDoubleBuffered(true);
        //--- --- --- --- --- ---
        //hitung numCarbon
        int numCarbon               = 0;
        this.nC                             = numCarbon;
        this.carbonX            = new Carbon[4];
        //--- --- --- --- --- ---
    }//end of canstructor

    Canvas(String sIsomer) {
        this.translateX     = 0;
        this.translateY     = 0;
        this.scale          = 1;
        this.setOpaque(false);
        this.setDoubleBuffered(true);
        //--- --- --- --- --- ---
        //hitung numCarbon
        int numCarbon               = 0;
        if((sIsomer!=null)&&(sIsomer.length()>1)){
            for(int i=1;i<sIsomer.length();i++){
                numCarbon   =
numCarbon+Integer.parseInt(String.valueOf(sIsomer.charAt(i)));
            }//end of for(int i=1;i<sIsomer.length();i++)
            if(Integer.parseInt(String.valueOf(sIsomer.charAt(0)))==1){
                numCarbon++;
            }//end                                                   of
if(numCarbon+Integer.parseInt(String.valueOf(sIsomer.charAt(0)))==1)
        }//end of if((sIsomer!=null)&&(sIsomer.length()>1))
```

```
            this.nC                             = numCarbon;
            Object[] address    = stringToArray(sIsomer);
            this.carbonX        = generateRantaiCarbon(address);
            //--- --- --- --- --- ---
    }//end of canstructor

    public void refresh(String sIsomer) {
            this.translateX     = 0;
            this.translateY     = 0;
            this.scale          = 1;
            this.setOpaque(false);
            this.setDoubleBuffered(true);
            //--- --- --- --- --- ---
            //hitung numCarbon
            int numCarbon               = 0;
            if((sIsomer!=null)&&(sIsomer.length()>1)){
                    for(int i=1;i<sIsomer.length();i++){
                            numCarbon   =
numCarbon+Integer.parseInt(String.valueOf(sIsomer.charAt(i)));
                    }//end of for(int i=1;i<sIsomer.length();i++)
                    if(Integer.parseInt(String.valueOf(sIsomer.charAt(0)))==1){
                            numCarbon++;
                    }//end                                                   of
if(numCarbon+Integer.parseInt(String.valueOf(sIsomer.charAt(0)))==1)
            }//end of if((sIsomer!=null)&&(sIsomer.length()>1))
            this.nC                             = numCarbon;
            Object[] address    = stringToArray(sIsomer);
            this.carbonX        = generateRantaiCarbon(address);
            //--- --- --- --- --- ---
            this.repaint();
    }//end of canstructor
```

```
    public void paint(Graphics g) {
            //-------------------------------------------------------------------------------------------------
--
            AffineTransform tx = new AffineTransform();
            tx.translate(translateX, translateY);
            tx.scale(scale, scale);
            Graphics2D graphAlkana2d = (Graphics2D) g;
            graphAlkana2d.setColor(Color.WHITE);
            ///graphAlkana2d.setComposite(AlphaComposite.SrcOver.derive(0.9F));
            graphAlkana2d.fillRect(0, 0, getWidth(), getHeight());
            graphAlkana2d.setTransform(tx);

    graphAlkana2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,Renderi
ngHints.VALUE_ANTIALIAS_ON);

    graphAlkana2d.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING,R
enderingHints.VALUE_TEXT_ANTIALIAS_ON);
            //-------------------------------------------------------------------------------------------------
--
            graphAlkana2d.setColor(Color.black);
            graphAlkana2d.setComposite(AlphaComposite.SrcOver.derive(0.1F));
            Font font       = new Font("Calibri", Font.BOLD, (this.gridSize));
            graphAlkana2d.setFont(font);
            String info1= "Isomer Alkana";
            if(this.nC>0){info1    = info1.concat(" C"+(this.nC)+"H"+(2*this.nC+2)+"");}
            graphAlkana2d.drawString(info1, 2*this.gridSize, 2*this.gridSize);
            int center1     = 0;
            if(carbonX[0]!=null){
                    center1         = carbonX[0].y;;
            }//end of if(carbonX[0]!=null)
```

```
            int lGrid       = 2*center1;
            lGrid           = lGrid*this.gridSize;
            float[] dash={4,2,4,2};
            int gridWidth= 0;
            Stroke                                                          stroke=new
BasicStroke(gridWidth,BasicStroke.CAP_BUTT,BasicStroke.JOIN_BEVEL,0,dash,0
);
            graphAlkana2d.setStroke(stroke);
            //draw poros and diagonal
            graphAlkana2d.setComposite(AlphaComposite.SrcOver.derive(0.3F));
            graphAlkana2d.setColor(Color.black);
            graphAlkana2d.drawLine(0, (lGrid/2), lGrid, (lGrid/2));
            graphAlkana2d.drawLine((lGrid/2), 0, (lGrid/2), lGrid);
            graphAlkana2d.drawLine(0, 0, lGrid, lGrid);
            graphAlkana2d.drawLine(0, lGrid, lGrid, 0);

            //draw border
            graphAlkana2d.setComposite(AlphaComposite.SrcOver.derive(0.4F));
            graphAlkana2d.setColor(Color.black);
            graphAlkana2d.drawLine(0, 0, lGrid, 0);
            graphAlkana2d.drawLine(0, 0, 0, lGrid);
            graphAlkana2d.drawLine(lGrid, 0, lGrid, lGrid);
            graphAlkana2d.drawLine(0, lGrid, lGrid, lGrid);
            //-----------------------------------------------------------------------------------------
--
            //preparing
            graphAlkana2d.setStroke(new BasicStroke(1));
            graphAlkana2d.setComposite(AlphaComposite.SrcOver.derive(1.0F));
            //untuk pohon bicetered

    if((this.carbonX[0]!=null)&&(this.carbonX[1]!=null)&&(this.carbonX[0].type==2)){
```

```
                int x0          = this.gridSize*this.carbonX[0].x;
                int y0          = this.gridSize*this.carbonX[0].y;
                int x1          = this.gridSize*this.carbonX[1].x;
                int y1          = this.gridSize*this.carbonX[1].y;
                graphAlkana2d.setColor(Color.gray);
                graphAlkana2d.drawLine(x0, y0, x1, y1);
                graphAlkana2d.setColor(Color.black);
                graphAlkana2d.fillOval(x0-6, y0-6, 12, 12);
                graphAlkana2d.fillOval(x0-6, y0-6, 12, 12);
        }//end                                                                   of
if((carbonX[0]!=null)&&(carbonX[1]!=null)&&(carbonX[0].type==2))
        //untuk pohon centered
        if((this.carbonX[0]!=null)&&(this.carbonX[0].type==1)){
                int center      = this.carbonX[0].y;
                int x0          = this.gridSize*center;
                int y0          = x0;
                int x1          = this.gridSize*this.carbonX[0].x;
                int y1          = this.gridSize*this.carbonX[0].y;
                graphAlkana2d.setColor(Color.gray);
                graphAlkana2d.drawLine(x0, y0, x1, y1);
                for(int i=0;i<this.carbonX.length;i++){
                        if(this.carbonX[i]!=null){
                                x1              = this.gridSize*this.carbonX[i].x;
                                y1              = this.gridSize*this.carbonX[i].y;
                                graphAlkana2d.setColor(Color.gray);
                                graphAlkana2d.drawLine(x0, y0, x1, y1);
                                graphAlkana2d.setColor(Color.black);
                                graphAlkana2d.fillOval(x0-6, y0-6, 12, 12);
                        }//end of if(this.carbonX[i]!=null)
                }//end of for(int i=0;i<this.carbonX.length;i++)
                //gambar lingkaran carbon
```

```
                graphAlkana2d.setColor(Color.black);
                graphAlkana2d.fillOval(x0-6, y0-6, 12, 12);
            }//end of if((carbonX[0]!=null)&&(carbonX[0].type==1))
            //melakuka penelusuran DFS untuk carbonX
            for(int i=0;i<carbonX.length;i++){
                ////System.out.println("Carbon["+(i)+"]");
                Carbon current      = carbonX[i];
                int dept            = 0;
                if(current!=null){
                    dept            = 1;
                    ////System.out.println("point:
("+(current.x)+","+(current.y)+")");
                }
                while((current!=null)&&(dept>0)){
                    int next        =0;
                    if(current.c1==null){
                        next    = -1;
                    }else{
                        if(current.c1.visited==false){
                            next    = 1;
                        }else{
                            if(current.c2==null){
                                next    = -1;
                            }else{
                                if(current.c2.visited==false){
                                    next    = 2;

                                }else{
                                    if(current.c3==null){
                                        next    = -1;
                                    }else{
```

```
    if(current.c3.visited==false){
                                                    next    = 3;

                                                }else{
                                                    next    = -1;
                                                }//end                   of
if(current.c3.visited==false)
                                                }//end of if(current.c1==null) else
                                            }//end of if(current.c2.visited==false)
                                        }//end of if(current.c2==null) else
                                    }//end of if(current.c1.visited==false)
                                }//end of if(current.c1==null) else
                                //System.out.println("next: "+(next));
                                //set next
                                int x0          = this.gridSize*current.x;
                                int y0          = this.gridSize*current.y;
                                if(next==-1){
                                    current.visited=true;
                                    dept--;
                                    if(current.c1!=null){current.c1.visited=false;}
                                    if(current.c2!=null){current.c2.visited=false;}
                                    if(current.c3!=null){current.c3.visited=false;}
                                    if(current.c0!=null){current  = current.c0;}
                                    graphAlkana2d.setColor(Color.black);
                                    graphAlkana2d.fillOval(x0-6, y0-6, 12, 12);
                                }else if(next==1){
                                    current         = current.c1;
                                    dept++;
                                    int x1          = this.gridSize*current.x;
                                    int y1          = this.gridSize*current.y;
```

```
                        graphAlkana2d.setColor(Color.gray);
                        graphAlkana2d.drawLine(x0, y0, x1, y1);
                    }else if(next==2){
                        current     = current.c2;
                        dept++;
                        int x1      = this.gridSize*current.x;
                        int y1      = this.gridSize*current.y;
                        graphAlkana2d.setColor(Color.gray);
                        graphAlkana2d.drawLine(x0, y0, x1, y1);
                    }else if(next==3){
                        current     = current.c3;
                        dept++;
                        int x1      = this.gridSize*current.x;
                        int y1      = this.gridSize*current.y;
                        graphAlkana2d.setColor(Color.gray);
                        graphAlkana2d.drawLine(x0, y0, x1, y1);
                    }
                }//end of while(dept>0)
            }//end of for(int i=0;i<carbonX.length;i++)
            //-------------------------------------------------------------------------------------------
--
            graphAlkana2d.dispose();
    }//end of paint

    public Object[]     stringToArray(String sIsomer){
            Object[] oAddress           = null;
            if((sIsomer!=null)&&(sIsomer.length()>1)){
                List<String> address    = new ArrayList();
                int type                =
Integer.parseInt(String.valueOf(sIsomer.charAt(0)));
```

```
                int value1              =
Integer.parseInt(String.valueOf(sIsomer.charAt(1)));
                String initialInformation      = ""+(value1)+""+(type)+"";
                address.add(initialInformation);
                int head0               = 1;
                int head1               = 2;
                int head2               = 0;//head untuk address
                int MAX_RADIUS    = 0;
                while(head1<sIsomer.length()){
                        if(sIsomer.charAt(head0)=='0'){
                                head0++;
                        }else{
                                if(head1<=head0){
                                        head1++;
                                }else{
                                        int interval     =
Integer.parseInt(String.valueOf(sIsomer.charAt(head0)));
                                        String address0       =
address.get(head2).substring(1,address.get(head2).length());
                                        for(int i=0;i<interval;i++){
                                                int value       =
Integer.parseInt(String.valueOf(sIsomer.charAt(head1)));
                                                if((value>0)&&(value<5)){
                                                        String address1       = ""+(1+i)+"";
                                                        String address2       =
""+(address0)+""+(address1)+"";

    address.add(""+(value)+""+(address2));
                                                        int newRadius         =
address2.length()-2;
                                                        if(MAX_RADIUS<newRadius){
```

```
                                                    MAX_RADIUS        =
newRadius;
                                                        }//end                          of
if(max_radian<(address2.length()-2))
                                                    }//end of if((value>0)&&(value<5))
                                                    head1++;
                                                    if(head1>=sIsomer.length()){break;}
                                                }//end of for(int i=0;i<interval;i++)
                                                head0++;
                                                head2++;
                                            }//end of if(head1<=head0) else
                                        }//end of if(sIsomer.charAt(head0)=='0') else

                        }//end of while(head1<sIsomer.length())
                        oAddress            = address.toArray();
                        oAddress[0]         = oAddress[0]+""+(2+MAX_RADIUS);
                }//end of if((sIsomer!=null)&&(sIsomer.length()>1))
                return oAddress;
        }//stringToList(String sIsomer)

        public Carbon[] generateRantaiCarbon(Object[] address){
                Carbon[] root = null;
                if((address!=null)&&(address.length>2)){
                        String initialInformation   = address[0].toString();
                        int numKuadran                      =
Integer.parseInt(String.valueOf(initialInformation.charAt(0)));
                        int type                            =
Integer.parseInt(String.valueOf(initialInformation.charAt(1)));
                        final int MAX_RADIUS            =
Integer.parseInt(initialInformation.substring(2, initialInformation.length()));
                        if(numKuadran>0){
```

```
root                                    = new Carbon[4];
root[0]                                 = null;
root[1]                                 = null;
root[2]                                 = null;
root[3]                                 = null;
final int center                = (int) (Math.floor((Math.pow(3.0,
MAX_RADIUS)+ 4.0)/2.0));
//System.out.println("center: "+(center));
int x1,y1,x2,y2,x3,y3,x4,y4;
x1                                              = center-2;
y1                                              = center;
x2                                              = center+2;
y2                                              = center;
x3                                              = center;
y3                                              = center-2;
x4                                              = center;
y4                                              = center+2;
int[]x                                  = {x1, x2, x3, x4};
int[]y                                  = {y1, y2, y3, y4};
String alamat1,alamat2,alamat3,alamat4;
alamat1                                         = "11";
alamat2                                         = "12";
alamat3                                         = "13";
alamat4                                         = "14";
if(type==2){
        alamat1                                 = "21";
        alamat2                                 = "22";
        alamat3                                 = "23";
        alamat4                                 = "24";
}//end of if(type==1) else
```

```
                String[]alamat = {alamat1, alamat2, alamat3, alamat4};
                for(int i=0;i<numKuadran;i++){
                    root[i] = new Carbon(alamat[i], x[i], y[i], MAX_RADIUS);
                }//end of for(int i=0;i<numKuadran;i++)
                //memulai proses pengisian Carbon;
                for(int i=1;i<address.length;i++){
                    String value = address[i].toString();
                    int numChild = Integer.parseInt(String.valueOf(value.charAt(0)));
                    String address1 = value.substring(1, value.length());
                    int index = (Integer.parseInt(String.valueOf(address1.charAt(1))))-1;
                    Carbon current = root[index];

                    ////System.out.println("trace:"+" address: "+(address1)+" - index: "+(index)+" - numChild: "+(numChild));
                    for(int j=2;j<address1.length();j++){
                        if(current!=null){
                            index = (Integer.parseInt(String.valueOf(address1.charAt(j))));
                            if(index==1){
                                current = current.c1;
                            }else if(index==2){
                                current = current.c2;
                            }else if(index==3){
                                current = current.c3;
                            }//end of if(index==1) else if
                        }else{
```

```
                                        break;
                                    }//end of if(current!=null) else
                                }//end of for(int j=1;j<address1.length();j++)
                                if(current!=null){
                                    current.addChild(numChild);

                                }//end of if(current!=null)
                                else{
                                    System.out.println("null");
                                }
                            }//end of for(int i=1;i<address.length;i++)
                        }//end of if(numKuadran>0)
                    }//end of if((address!=null)&&(address.length>2))
                    return root;
                }//end of generateRantaiCarbon(Object[] address)

            }//end of class Canvas
```

8. IsomerAlkana.java

```
package com.alakana.isomer;

import javax.swing.SwingUtilities;
import java.awt.BorderLayout;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.awt.Dimension;
import java.awt.GridBagLayout;
import java.awt.FlowLayout;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import java.awt.event.MouseWheelEvent;
import java.awt.event.MouseWheelListener;
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.SwingConstants;
import javax.swing.JTabbedPane;
import javax.swing.JScrollPane;
```

```
import javax.swing.JTextArea;

public class IsomerAlkana extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel jContentPane = null;
    private JPanel jPanelNort = null;
    private JPanel jPanelCenter = null;
    private JLabel jLabelNumC = null;
    private JTextField jTextFieldNumC = null;
    private JButton jButtonRunValidator = null;
    private JButton jButtonShow = null;
    private JLabel jLabelNumIsomer = null;
    private JTextField jTextFieldNumIsomer = null;
    private JButton jButtonPrev = null;
    private JTextField jTextFieldIndex = null;
    private JButton jButtonNext = null;
    private JTabbedPane jTabbedPane = null;
    private JPanel jPanelPreview = null;
    private JPanel jPanelLog = null;
    private static Canvas canvas    = null;
    private JScrollPane jScrollPane = null;
    private JTextArea jTextAreaLog = null;
    private int indexCarbon                    = 0;
    private int MaxIndex                = 0;
    private String[] address            = null;

    /**
     * This method initializes jPanelNort
     *
     * @return javax.swing.JPanel
```

```
 */
private JPanel getJPanelNort() {
	if (jPanelNort == null) {
		jLabelNumIsomer = new JLabel();
		jLabelNumIsomer.setText("Isomer");
		jLabelNumIsomer.setHorizontalAlignment(SwingConstants.RIGHT);
		jLabelNumIsomer.setPreferredSize(new Dimension(60, 30));
		jLabelNumC = new JLabel();
		jLabelNumC.setText("Carbon: ");
		jLabelNumC.setPreferredSize(new Dimension(60, 30));
		jPanelNort = new JPanel();
		jPanelNort.setLayout(new FlowLayout());
		jPanelNort.add(jLabelNumC, null);
		jPanelNort.add(getJTextFieldNumC(), null);
		jPanelNort.add(getJButtonRunValidator(), null);
		jPanelNort.add(jLabelNumIsomer, null);
		jPanelNort.add(getJTextFieldNumIsomer(), null);
		jPanelNort.add(getJButtonShow(), null);
		jPanelNort.add(getJButtonPrev(), null);
		jPanelNort.add(getJTextFieldIndex(), null);
		jPanelNort.add(getJButtonNext(), null);
	}
	return jPanelNort;
}

/**
 * This method initializes jPanelCenter
 *
 * @return javax.swing.JPanel
 */
private JPanel getJPanelCenter() {
```

```
		if (jPanelCenter == null) {
			jPanelCenter = new JPanel();
			jPanelCenter.setLayout(new BorderLayout());
			jPanelCenter.add(getJTabbedPane(), BorderLayout.CENTER);
		}
		return jPanelCenter;
	}

	/**
	 * This method initializes jTextFieldNumC
	 *
	 * @return javax.swing.JTextField
	 */
	private JTextField getJTextFieldNumC() {
		if (jTextFieldNumC == null) {
			jTextFieldNumC = new JTextField();
			jTextFieldNumC.setPreferredSize(new Dimension(100, 30));
			jTextFieldNumC.setHorizontalAlignment(JTextField.RIGHT);
		}
		return jTextFieldNumC;
	}

	/**
	 * This method initializes jButtonRunValidator
	 *
	 * @return javax.swing.JButton
	 */
	private JButton getJButtonRunValidator() {
		if (jButtonRunValidator == null) {
			jButtonRunValidator = new JButton();
			jButtonRunValidator.setText("Hitung Isomer");
```

```
                jButtonRunValidator.setPreferredSize(new Dimension(150, 30));
                jButtonRunValidator.addActionListener(new
java.awt.event.ActionListener() {
                        public void actionPerformed(java.awt.event.ActionEvent e) {
                                int nCarbon   =
Integer.parseInt(jTextFieldNumC.getText().trim());
                                runValidator(nCarbon);
                                System.out.println("actionPerformed()              Hitung
Isomer"); // TODO Auto-generated Event stub actionPerformed()
                        }
                });
        }
        return jButtonRunValidator;
    }

    /**
     * This method initializes jButtonShow
     *
     * @return javax.swing.JButton
     */
    private JButton getJButtonShow() {
        if (jButtonShow == null) {
                jButtonShow = new JButton();
                jButtonShow.setText("Konstruksi Isomer");
                jButtonShow.setPreferredSize(new Dimension(150, 30));
                jButtonShow.addActionListener(new
java.awt.event.ActionListener() {
                        public void actionPerformed(java.awt.event.ActionEvent e) {
                                konstruksiIsomer();
                                System.out.println("actionPerformed()");     //     TODO
Auto-generated Event stub actionPerformed()
```

```
                        }
                });
        }
        return jButtonShow;
}

/**
 * This method initializes jTextFieldNumIsomer
 *
 * @return javax.swing.JTextField
 */
private JTextField getJTextFieldNumIsomer() {
        if (jTextFieldNumIsomer == null) {
                jTextFieldNumIsomer = new JTextField();
                jTextFieldNumIsomer.setPreferredSize(new Dimension(100, 30));
                jTextFieldNumIsomer.setHorizontalAlignment(JTextField.RIGHT);
        }
        return jTextFieldNumIsomer;
}

/**
 * This method initializes jButtonPrev
 *
 * @return javax.swing.JButton
 */
private JButton getJButtonPrev() {
        if (jButtonPrev == null) {
                jButtonPrev = new JButton();
                jButtonPrev.setText("Prev");
                jButtonPrev.setPreferredSize(new Dimension(60, 30));
```

```
                jButtonPrev.addActionListener(new   java.awt.event.ActionListener()
{
                        public void actionPerformed(java.awt.event.ActionEvent e) {
                                int index       =
Integer.parseInt(jTextFieldIndex.getText().trim());
                                indexCarbon = index;
                                if((address!=null)&&(address.length>index)){
                                        canvas.refresh(address[index]);
                                }
                                int newIndex = -1+index;
                                if(newIndex<0){
                                        newIndex=-1+address.length;
                                }
                                jTextFieldIndex.setText(""+(newIndex));
                                System.out.println("actionPerformed()");     //     TODO
Auto-generated Event stub actionPerformed()
                        }
                });
        }
        return jButtonPrev;
    }

    /**
     * This method initializes jTextFieldIndex
     *
     * @return javax.swing.JTextField
     */
    private JTextField getJTextFieldIndex() {
        if (jTextFieldIndex == null) {
                jTextFieldIndex = new JTextField();
                jTextFieldIndex.setPreferredSize(new Dimension(100, 30));
```

```
                jTextFieldIndex.setText("0");
            }
            return jTextFieldIndex;
    }

    /**
     * This method initializes jButtonNext
     *
     * @return javax.swing.JButton
     */
    private JButton getJButtonNext() {
            if (jButtonNext == null) {
                jButtonNext = new JButton();
                jButtonNext.setText("Next");
                jButtonNext.setPreferredSize(new Dimension(100, 30));
                jButtonNext.addActionListener(new   java.awt.event.ActionListener()
{
                        public void actionPerformed(java.awt.event.ActionEvent e) {
                            int index        =
Integer.parseInt(jTextFieldIndex.getText().trim());
                            indexCarbon = index;
                            if((address!=null)&&(address.length>index)){
                                canvas.refresh(address[index]);
                            }
                            int newIndex = 1+index;
                            if(newIndex>address.length){
                                newIndex=0;
                            }
                            jTextFieldIndex.setText(""+(newIndex));
```

```
                                        System.out.println("actionPerformed()");      //      TODO
Auto-generated Event stub actionPerformed()
                                }
                        });
                }
                return jButtonNext;
        }

        /**
         * This method initializes jTabbedPane
         *
         * @return javax.swing.JTabbedPane
         */
        private JTabbedPane getJTabbedPane() {
                if (jTabbedPane == null) {
                        jTabbedPane = new JTabbedPane();
                        jTabbedPane.setTabPlacement(JTabbedPane.BOTTOM);
                        jTabbedPane.addTab("Preview", null, getJPanelPreview(), null);
                        jTabbedPane.addTab("Log", null, getJPanelLog(), null);
                }
                return jTabbedPane;
        }

        /**
         * This method initializes jPanelPreview
         *
         * @return javax.swing.JPanel
         */
        private JPanel getJPanelPreview() {
                if (jPanelPreview == null) {
                        jPanelPreview = new JPanel();
```

```
                jPanelPreview.setLayout(new BorderLayout());
                jPanelPreview.add(getCanvas(), BorderLayout.CENTER);//
            }
            return jPanelPreview;
    }

    /**
     * This method initializes jPanelLog
     *
     * @return javax.swing.JPanel
     */
    private JPanel getJPanelLog() {
            if (jPanelLog == null) {
                jPanelLog = new JPanel();
                jPanelLog.setLayout(new BorderLayout());
                jPanelLog.add(getJScrollPane(), BorderLayout.CENTER);
            }
            return jPanelLog;
    }

    /**
     * This method initializes canvas
     *
     * @return Canvas
     */
    private Canvas getCanvas(){
            if(canvas==null){
                canvas                                  =                                  new
Canvas();//("2221201");//("143333333333333333");//("123200000");
            KeyHandler keyHandler    = new KeyHandler();
            this.addKeyListener(keyHandler);
```

```
        TranslateHandler translateHandler = new TranslateHandler();
        canvas.addMouseListener(translateHandler);
        canvas.addMouseMotionListener(translateHandler);
        ScaleHandler scaleHandler    = new ScaleHandler();
        canvas.addMouseWheelListener(scaleHandler);
          }
          return canvas;
}

/**
 *
 * @param nCarbon
 */
private void runValidator(int nCarbon){
          jTextAreaLog.append("--- --- --- --- --- --- --- --- --- --- ---\n");
          jTextAreaLog.append("C"+(nCarbon)+"H"+(2*nCarbon+2)+"\n");
          if(nCarbon<5){
                    jTextAreaLog.append("jumlah karbon harus>4\n");
                    jTextAreaLog.append("--- --- --- --- --- --- --- --- --- --- ---\n");
          }else{
                    jTextAreaLog.append("Memulai Proses Count Isomer...\n");
                    int MAX                 = nCarbon;
                    int[][]arrT             = new int[MAX][];
                    int[][]arrS2    = new int[MAX][];
                    int[][]arrS3    = new int[MAX][];
                    int[][]arrS4    = new int[MAX][];
                    int[][]arrB             = new int[MAX][];
                    int[][]arrC             = new int[MAX][];


                    int[]T_1                = {1};
```

```
int[]T0                 = {1,1};
arrT[0]                 = T_1.clone();
arrT[1]                 = T0.clone();

S3 s3T0                         = new S3(arrT[1].clone());
arrS3[1]                = s3T0.result.clone();

S4 s4T_1                = new S4(arrT[0].clone());
arrS4[0]                = s4T_1.result.clone();

S4 s4T0                         = new S4(arrT[1].clone());
arrS4[1]                = s4T0.result.clone();

int[]c0                 = {0,1};
arrC[1]                         = c0.clone();

//System.out.print("Hitung T");
jTextAreaLog.append("do proses");
for(int i=2;i<arrT.length;i++){
        //hitung T1 dst
        PolynomialZet pz1   = new PolynomialZet();
        pz1.insert(arrS3[-1+i].clone());
        int[]z0={0,1};
        pz1.times(z0);
        int[]z1={1};
        pz1.plus(z1);
        arrT[i] = pz1.getZet().clone();

        //hitung S3T1 dst
        S3 s3  = new S3(arrT[i].clone());
        arrS3[i]=s3.result.clone();
```

```
//hitung S4T1 dst
S4 s4  = new S4(arrT[i].clone());
arrS4[i]= s4.result.clone();

//hitung C
PolynomialZet pz2   = new PolynomialZet();
pz2.insert(arrS4[-1+i].clone());
pz2.times(z0);
pz2.plus(z1);

PolynomialZet pz3   = new PolynomialZet();
pz3.insert(arrS4[-2+i].clone());
pz3.times(z0);
pz3.plus(z1);
pz2.minus(pz3.getZet());

PolynomialZet pz4   = new PolynomialZet();
pz4.insert(arrT[-1+i].clone());
pz4.minus(arrT[-2+i].clone());

PolynomialZet pz5   = new PolynomialZet();
pz5.insert(arrT[-1+i].clone());
pz5.minus(z1);
pz4.times(pz5.getZet());
pz2.minus(pz4.getZet());
arrC[i] = pz2.getZet().clone();

//hitung B
PolynomialZet pz6   = new PolynomialZet();
pz6.insert(arrT[-1+i].clone());
```

```
            pz6.minus(arrT[-2+i].clone());
            S2 s2  = new S2(pz6.getZet());
            arrB[i] = s2.result.clone();

            //System.out.print(".");
            jTextAreaLog.append(".");
    }//end of for i
    //System.out.println("[OK]");
    jTextAreaLog.append("[OK]\n");

    //hitung Sum C
    PolynomialZet pzC  = new PolynomialZet();
    pzC.insert(arrC[1].clone());

    PolynomialZet pzB  = new PolynomialZet();

    PolynomialZet pzIso          = new PolynomialZet();

    //System.out.print("Hitung C dan B");
    jTextAreaLog.append("Hitung Centered dan Bicentered");
    for(int i=2;i<arrC.length;i++){
            pzC.plus(arrC[i]);
            pzB.plus(arrB[i]);
            //System.out.print(".");
            jTextAreaLog.append(".");
    }
    //System.out.println("[OK]");
    jTextAreaLog.append("[OK]\n");

    pzIso.plus(pzC.getZet().clone());
    pzIso.plus(pzB.getZet().clone());
```

```
                    //System.out.println("Sigma_C: "+(pzC.toString()));
                    //System.out.println("Sigma_B: "+(pzB.toString()));
                    //System.out.println("Sigma_BC: "+(pzIso.toString()));
                    jTextAreaLog.append("Sigma_C : "+(pzC.toString())+"[OK]\n");
                    jTextAreaLog.append("Sigma_B : "+(pzB.toString())+"[OK]\n");
                    jTextAreaLog.append("Sigma_BC: "+(pzIso.toString())+"[OK]\n");
                    jTextAreaLog.append("--- --- --- --- --- --- --- --- --- --- ---\n");
                    int nIso        = pzIso.getValueAt(nCarbon);
                    MaxIndex    = nIso;
                    jTextFieldNumIsomer.setText(""+(nIso)+"");
                    jTextAreaLog.append("Jumlah Isomer Alkana Untuk
C"+(nCarbon)+"H"+(2*nCarbon+2)+" adalah: "+(nIso)+"\n\n\n");
            }//end of if(nCarbon<4) else
    }//end of runValidator(int nCarbon)

    /**
     *
     * @param c
     * @return
     */
    private Object[] generateArray(int c){
            Object[] address    = null;
            String filename     ="result/C"+(c)+"H"+(2*c+2)+".txt";
            FileInputStream fis;
            InputStreamReader isr;
            BufferedReader br;
            try{
                    fis     = new FileInputStream(filename);
                    isr     = new InputStreamReader(fis);
                    br      = new BufferedReader(isr);
```

```
                List<String> lista      = new ArrayList();
                String thisLine;
                while((thisLine=br.readLine())!=null){
                        lista.add(thisLine);
                }//end of while((thisLine=br.readLine())!=null)
                address         = lista.toArray();
        }catch(Exception e){
                e.printStackTrace();
        }
        return address;
    }//end of generateArray

    private void konstruksiIsomer(){
        int c    = Integer.parseInt(jTextFieldNumC.getText().trim());
        Object[] oAddress    = generateArray(c);
        if(oAddress!=null){
                jTextAreaLog.append("--- --- --- --- --- --- --- --- --- --- ---\n");
                jTextAreaLog.append("Rumus Struktur Pohon Centered dan
Bicentered\n");
                this.address  = new String[oAddress.length];
                for(int i=0;i<this.address.length;i++){
                        this.address[i]         = oAddress[i].toString();
                        jTextAreaLog.append("-"+(this.address[i])+"\n");
                }
                jTextFieldNumIsomer.setText(""+(oAddress.length));
                jTextAreaLog.append("--- --- --- --- --- --- --- --- --- --- ---\n\n\n");
                //System.out.println("oAddress.length: "+(oAddress.length));
        }//end of if(oAddress!=null)
    }//end of konstruksiIsomer()
```

```
/**
 * This method initializes jScrollPane
 *
 * @return javax.swing.JScrollPane
 */
private JScrollPane getJScrollPane() {
	if (jScrollPane == null) {
		jScrollPane = new JScrollPane();
		jScrollPane.setViewportView(getJTextAreaLog());
	}
	return jScrollPane;
}

/**
 * This method initializes jTextAreaLog
 *
 * @return javax.swing.JTextArea
 */
private JTextArea getJTextAreaLog() {
	if (jTextAreaLog == null) {
		jTextAreaLog = new JTextArea();
	}
	return jTextAreaLog;
}

/**
 * @param args
 */
public static void main(String[] args) {
	// TODO Auto-generated method stub
	SwingUtilities.invokeLater(new Runnable() {
```

```
                public void run() {
                        IsomerAlkana thisClass = new IsomerAlkana();

thisClass.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                        thisClass.setVisible(true);
                }
        });
}

private static class KeyHandler implements KeyListener{

        @Override
        public void keyPressed(KeyEvent arg0) {
                // TODO Auto-generated method stub

        }

        @Override
        public void keyReleased(KeyEvent arg0) {
                // TODO Auto-generated method stub

        }

        @Override
        public void keyTyped(KeyEvent arg0) {
                // TODO Auto-generated method stub

        }

}
```

```
    private static class TranslateHandler implements MouseListener, MouseMotionListener {

        private int lastOffsetX;
    private int lastOffsetY;

        @Override
        public void mouseDragged(MouseEvent e) {
            // new x and y are defined by current mouse location subtracted
       // by previously processed mouse location
            int newX = e.getX() - lastOffsetX;
            int newY = e.getY() - lastOffsetY;

            // increment last offset to last processed by drag event.
            lastOffsetX += newX;
            lastOffsetY += newY;

            // update the canvas locations
            canvas.translateX += newX;
            canvas.translateY += newY;

            // schedule a repaint.
            canvas.repaint();

        }

        @Override
        public void mouseMoved(MouseEvent arg0) {
            // TODO Auto-generated method stub

        }
```

```
    @Override
    public void mouseClicked(MouseEvent arg0) {
            // TODO Auto-generated method stub

    }

    @Override
    public void mouseEntered(MouseEvent arg0) {
            // TODO Auto-generated method stub

    }

    @Override
    public void mouseExited(MouseEvent arg0) {
            // TODO Auto-generated method stub

    }

    @Override
    public void mousePressed(MouseEvent e) {
            // mengcapture starting point
lastOffsetX = e.getX();
lastOffsetY = e.getY();

    }

    @Override
    public void mouseReleased(MouseEvent arg0) {
            // TODO Auto-generated method stub
```

```
            }

    }

    private static class ScaleHandler implements MouseWheelListener {

            @Override
            public void mouseWheelMoved(MouseWheelEvent e) {
                    if(e.getScrollType()                                    ==
MouseWheelEvent.WHEEL_UNIT_SCROLL) {
             canvas.scale += (.1 * e.getWheelRotation());
             canvas.scale = Math.max(0.00001, canvas.scale);
             canvas.repaint();
           }
            }

    }

    /**
     * This is the default constructor
     */
    public IsomerAlkana() {
            super();
            initialize();
    }

    /**
     * This method initializes this
     *
     * @return void
     */
```

```
    private void initialize() {
            try {

UIManager.setLookAndFeel(ch.randelshofer.quaqua.QuaquaManager.getLookAnd
Feel());
        // set UI manager properties here that affect Quaqua
            } catch (Exception e1) {
                    try{

    UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
                            }
                            catch (Exception e3) {
                                    // handle exception
                            }
                    System.out.println("L&F Quaqua gagal");
            }//end of set Look And Feel
            this.setSize(1024, 540);
            this.setContentPane(getJContentPane());
            this.setTitle("Isomer Alkana");
    }

    /**
     * This method initializes jContentPane
     *
     * @return javax.swing.JPanel
     */
    private JPanel getJContentPane() {
            if (jContentPane == null) {
                    jContentPane = new JPanel();
                    jContentPane.setLayout(new BorderLayout());
                    jContentPane.add(getJPanelNort(), BorderLayout.NORTH);
```

```
                jContentPane.add(getJPanelCenter(), BorderLayout.CENTER);
            }
            return jContentPane;
    }

}
```