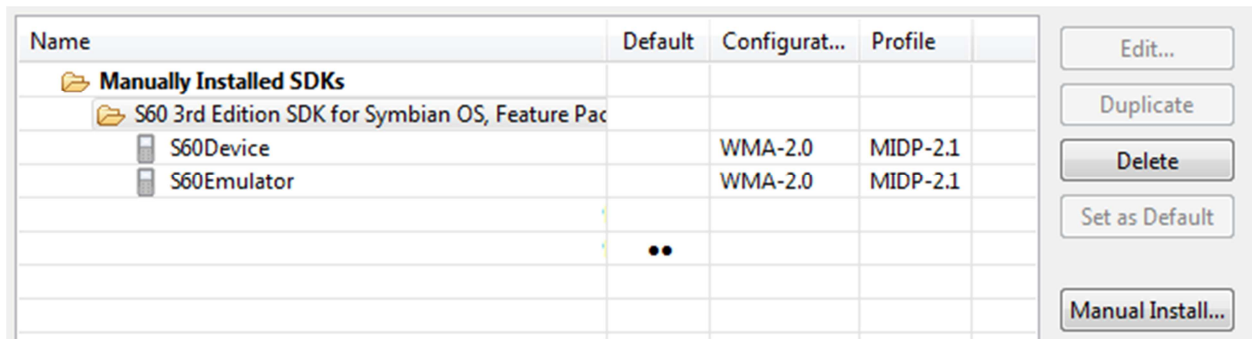


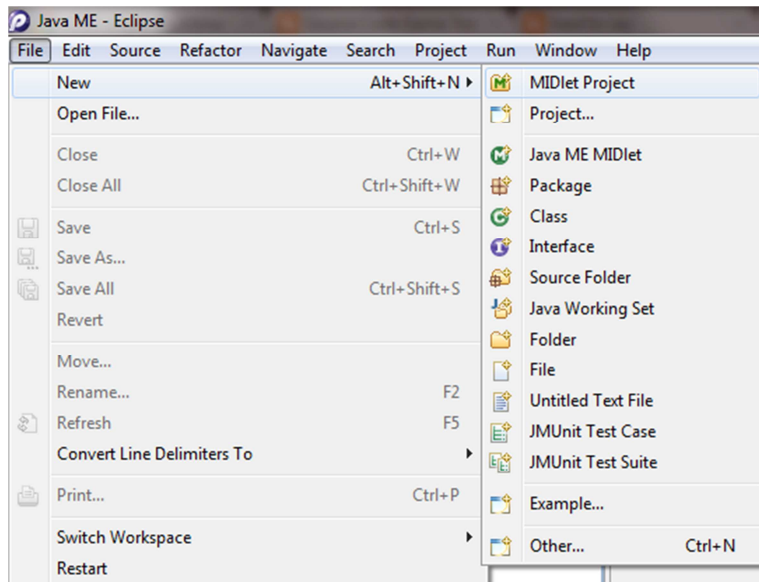
Browse ke folder dimana kita menyimpan Emulatornya. Lalu tekan finish, tunggu beberapa saat, nanti akan muncul seperti ini.



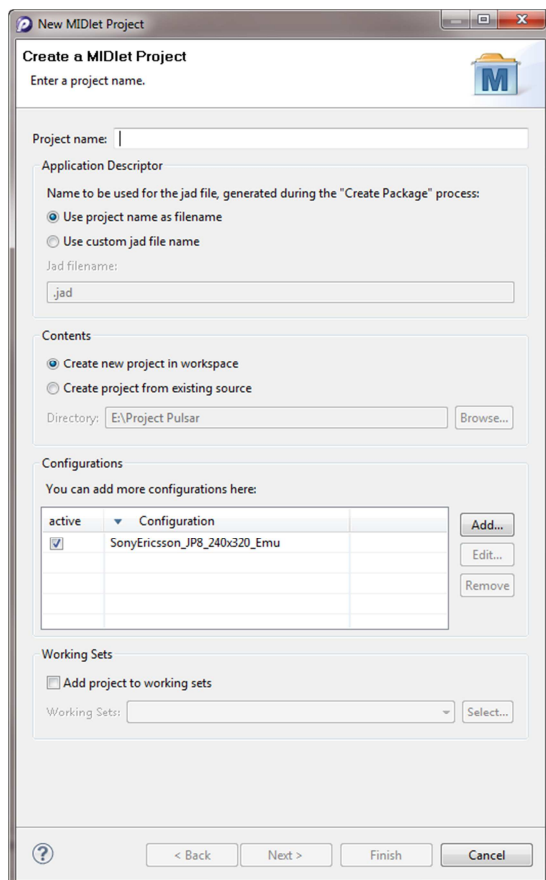
Lalu tekan OK untuk menutup Preferences

#3 Membuat MIDlet Project

Membuat MIDlet Project bisa dilakukan dengan cara seperti ini:

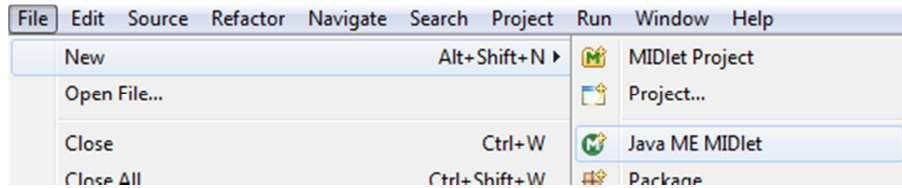


Lalu isi detail dari project kita nanti

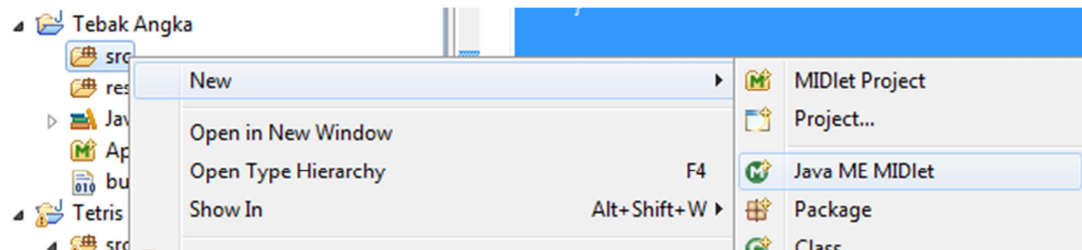


#4 Membuat file MIDlet

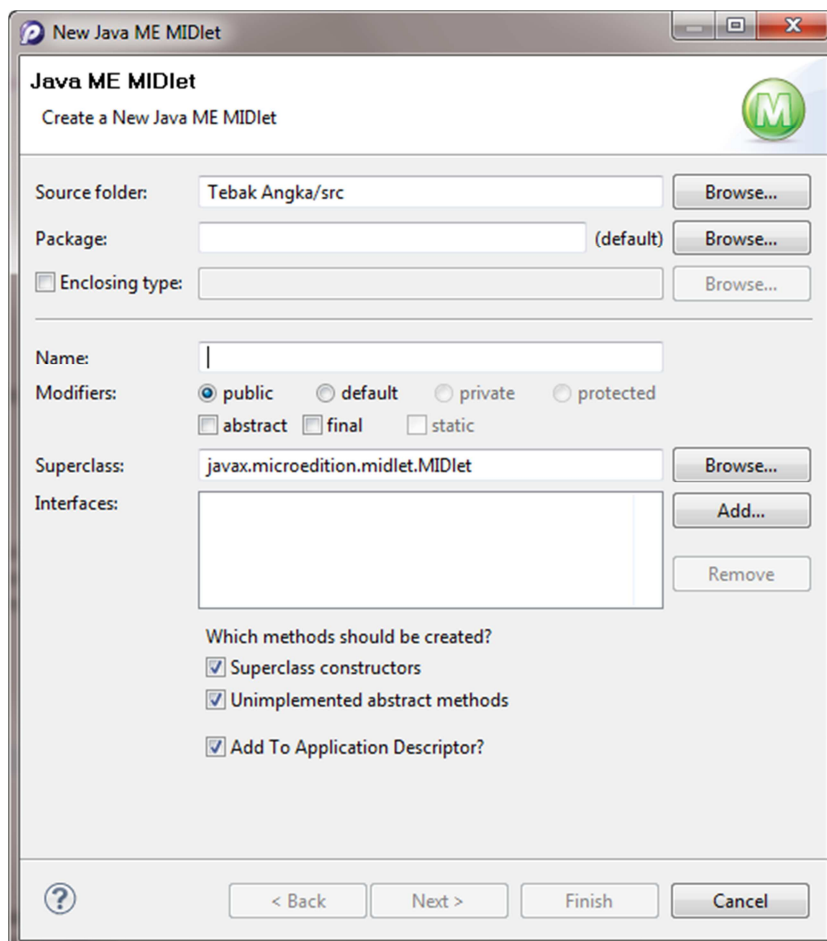
Membuat file MIDlet bisa dilakukan dengan 2 cara, yaitu:



atau



Isi detail dari file MIDlet kita:



Nanti akan muncul file kita yang isinya:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class TebakAngka extends MIDlet{
    public TebakAngka () {
    }

    public void startApp () {
    }

    public void pauseApp () { }

    public void destroyApp (boolean b) {
    }
}
```

Ketika kita membuat file MIDlet, akan selalu terbentuk 3 buah method seperti diatas. Method yang pertama adalah startApp() dipanggil oleh Application Management Software (AMS) ketika MIDlet dijalankan atau diresume. Resume, dengan kata lain startApp() akan kembali dijalankan setelah di-pause. Maka berhati-hatilah ketika kita menempatkan inisialisasi di dalam startApp().

Method yang kedua adalah pauseApp(), yang akan di-invoke oleh AMS ketika dalam situasi tertentu, seperti pengguna menerima telepon ketika bermain game, peringatan low battery, dan lain-lain. Dan dapat digunakan untuk meminimasi CPU, memory, atau konsumsi baterenya, ketika game kita tidak dalam active state.

Method yang terakhir adalah destroyApp(). Method ini dipanggil untuk destroy MIDlet oleh AMS. Dalam method ini harus clean-up resource, action, dan lain-lain. Dan memberitahukan AMS untuk menyempurnakan cleanup dengan notifDestroy()

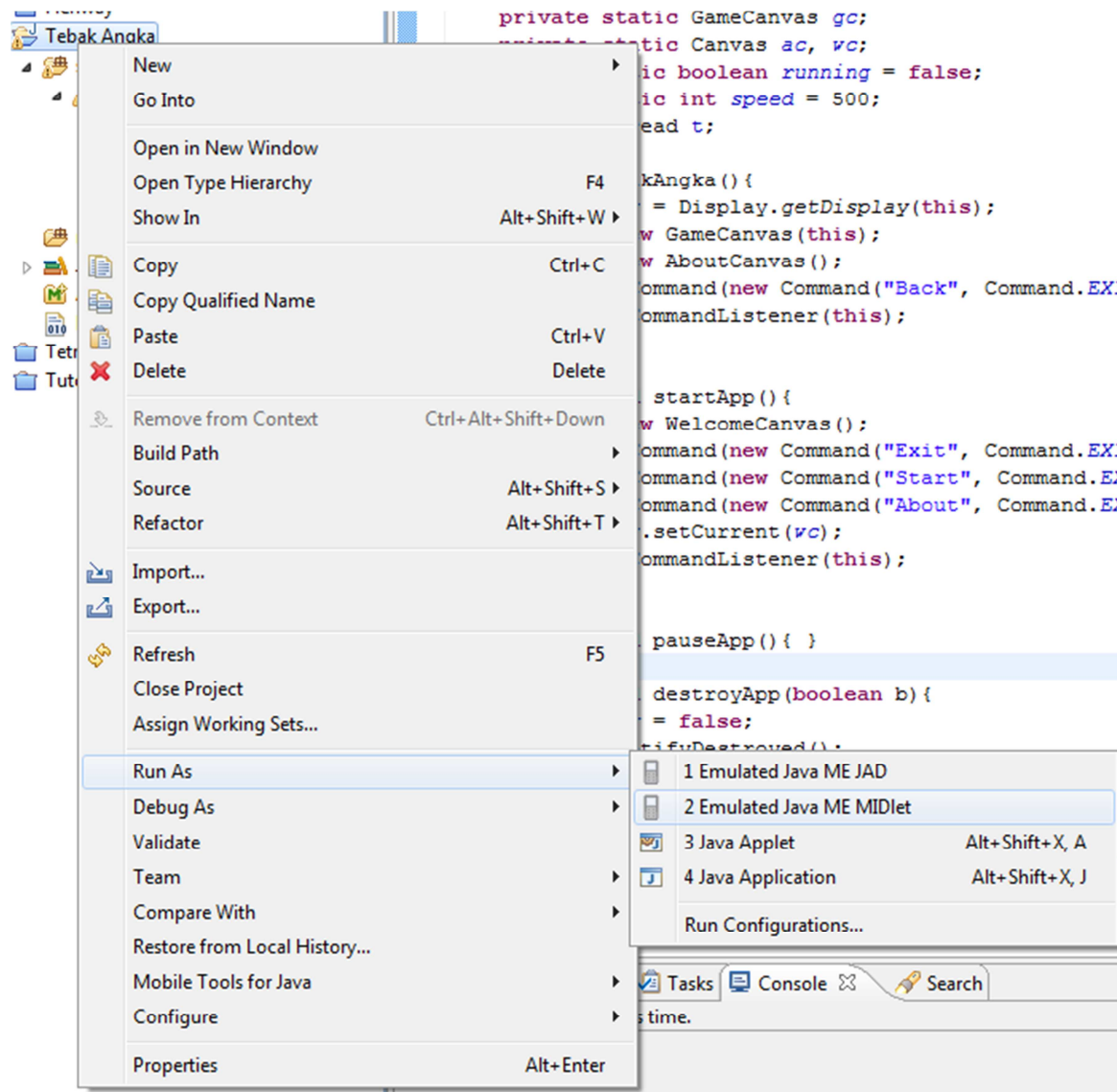
Ketiga method di atas merupakan bagian dari MIDP Lifecycle. Sangat penting untuk semua programmer J2ME memahaminya.

#Membuat Hello World

ketik di fungsi startApp()

```
System.out.println("Hello World...");
```

Lalu jalankan emulator seperti dibawah ini:

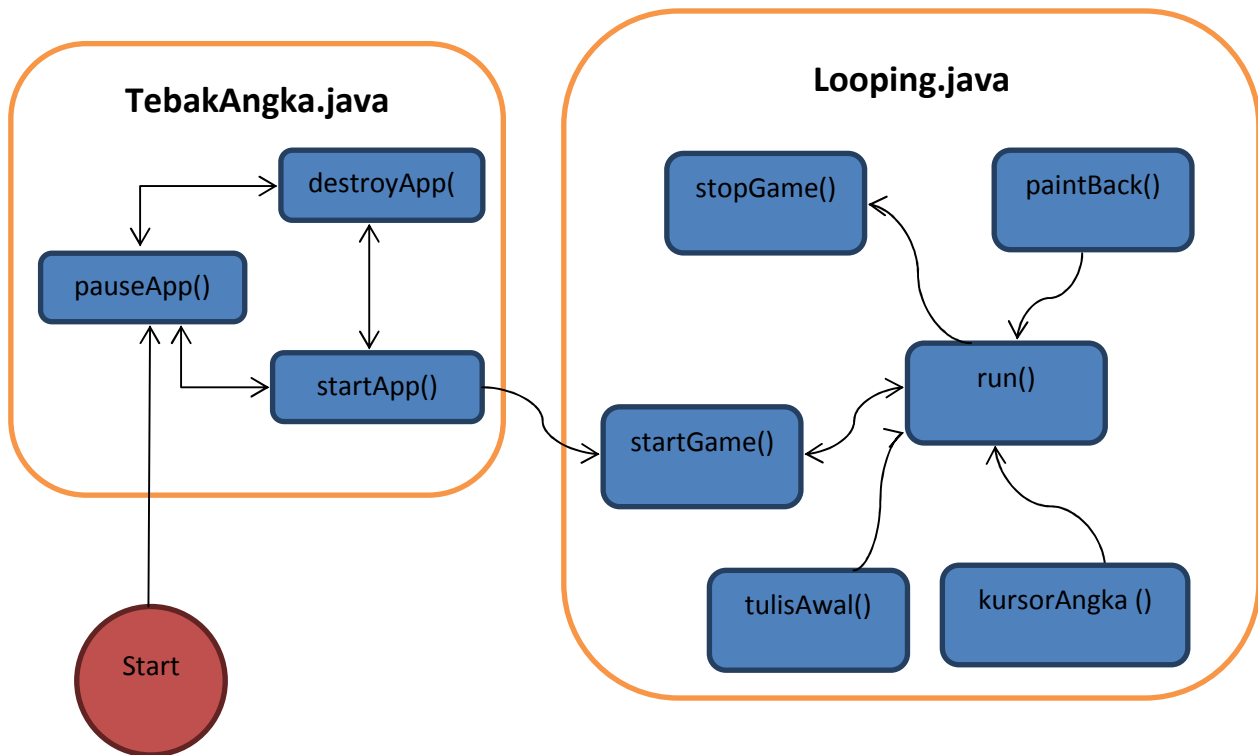


#Membuat game Tebak Angka

Rules:

- Random angka dari 1-20
- Ketika kita menebak angka, akan ada penjelasan bahwa angka yang kita tebak itu terlalu kecil atau terlalu besar.
- Jika benar, maka permainan berhenti.

Struktur game:



Penjelasan:

TebakAngka.java

- Variable yang digunakan yaitu,

```
private final static int PAUSED = 0;
private final static int ACTIVE = 1;
private final static int DESTROYED = 2;
private static int stateTracker = PAUSED;

Command exitCommand = new Command("Exit", Command.EXIT, 2);

private Display gameDisplay;
private Looping gameloop;
```

- startApp(), berfungsi untuk menampilkan Display dan memuat game canvas kedalam display

```
public void startApp() {
    if (gameloop == null)
    {
        gameloop = new Looping(this);
        gameloop.startGame();

        //add some command
        gameloop.addCommand(exitCommand);
    }
}
```

```
        gameloop.setCommandListener(this);  
    }  
    gameDisplay.getDisplay(this).setCurrent(gameloop);  
    stateTracker = ACTIVE;  
}
```

- pauseApp(), untuk memanggil notifyPaused()

```
public void pauseApp()  
{  
    stateTracker = PAUSED;  
    notifyPaused();  
}
```

- destroyApp() dan performDestroyCleanup(), untuk menutup aplikasi

```
private synchronized void performDestroyCleanup() {  
    if (stateTracker != DESTROYED) {  
        stateTracker = DESTROYED;  
        notifyDestroyed();  
    }  
}  
  
public void destroyApp(boolean b) throws  
MIDletStateChangeException {  
    performDestroyCleanup();  
}
```

Looping.java

- Variable yang digunakan yaitu,

```
Graphics g;  
private int kursorMenu=1;  
private int inputDelay = 0;  
private int angka = 0;  
Random acak = new Random();  
  
private MIDlet parent;  
  
public long GameFrameDelay = 30;  
Thread thread = new Thread(this);
```

- Konstruktor dari Looping.java

```
protected Looping(MIDlet parent) {  
    super(true);  
    this.setFullScreenMode(true);  
    g = getGraphics();  
    this.parent = parent;  
}
```

Disini kita membuat game kita fullscreen mode,
dengan menambahkan `this.setFullScreenMode(true);`

Lalu `g = getGraphics()` berarti kita menginstansiasikan `g`.

- `startGame()`, untuk menjalankan Thread.

```
publicvoid startGame() {  
    thread.start();  
}
```

- `stopGame()`, untuk memberhentikan Thread.

```
publicvoid stopGame() {  
    thread = null;  
}
```

- `paintBack()`, untuk menggambar background game.

```
publicvoid paintBack() {  
    g.setColor(139, 139, 139);  
    g.fillRect(0, 0, getWidth(), getHeight());  
}
```

- `tulisAwal()`, untuk menampilkan tulisan di game.

```
publicvoid tulisAwal() {  
    g.setColor(0, 0, 0);  
    g.drawString("Masukan nomornya:", 0, 0, 0);  
}
```

- `kursorAngka()`, untuk menghandle input user, ada angka di tengah layar, ketika kita tekan atas maka angka akan bertambah, begitu pula sebaliknya. Jika tombol FIRE ditekan, maka akan terjadi pengecekan apakah angka yang kita pilih itu lebih besar, lebih kecil, atau benar.

```
publicvoid kursorAngka(int angka){  
    g.setColor(255, 255, 255);  
    g.drawString(""+kursorMenu, getWidth()/2, getHeight()/3, 0);  
  
    if (++inputDelay > 2) {  
        int keyState = getKeyStates();  
  
        if((keyState &DOWN_PRESSED) != 0){  
            if (kursorMenu == 1) {  
                kursorMenu = 20;  
            }else {  
                kursorMenu--=1;  
            }  
        }  
  
        if((keyState &UP_PRESSED) != 0){  
            if (kursorMenu == 20) {  
                kursorMenu = 1;  
            }else {  
                kursorMenu+=1;  
            }  
        }  
    }  
}
```

```
    }  
  
    if((keyState &FIRE_PRESSED) != 0){  
        System.out.println("Anda menjawab : " + kursorMenu);  
        if(kursorMenu< angka)  
            System.out.println("Tebakan terlalu kecil!");  
        elseif(kursorMenu> angka)  
            System.out.println("Tebakan terlalu besar");  
        else  
        {  
            System.out.println("Benar");  
            parent.notifyDestroyed();  
        }  
    }  
    inputDelay = 0;  
}  
}
```

Disini kita akan menampilkan angka ditengah layar dengan menggunakan drawstring. Kemudian fungsi pengecekan inputDelay itu untuk menahan agar ketika kita menekan suatu tombol, maka tidak terlalu cepat geraknya. Lalu kita tambahkan lagi fungsi IF untuk mengecek apakah tombol bawah atau atas ditekan. Ketika tombol atas ditekan, maka ada cek ulang, apakah kursorMenu == 20, jika iya, maka kursorMenu telah mencapai batas paling atas yang telah kita tentukan sebelumnya. Maka kita set ulang agar kursorMenu berubah kembali menjadi 1. ELSE, kursorMenu akan terus bertambah nilainya. Begitu pula sebaliknya jika tombol bawah ditekan. Lalu terakhir ada pengecekan apakah tombol FIRE ditekan, jika iya, maka ada pengecekan apakah nilai yang Anda pilih (kursorMenu) terlalu besar atau terlalu kecil. Jika benar maka panggil fungsi notifyDestroyed().

- run(), untuk menyimpan gameloop kita. Di dalam game loop, kita taruh juga fungsi-fungsi lain yang akan dieksekusi terus menerus, yaitu tulisAwal(), paintBack(), dan kursorAngka().

```
public void run() {  
    // TODO Auto-generated method stub  
    angka = acak.nextInt(19)+1;  
    System.out.println(angka);  
    while (true) {  
        paintBack();  
        tulisAwal();  
        kursorAngka(angka);  
        try {  
            Thread.sleep(GameFrameDelay);  
        } catch (InterruptedException ie) {  
            stopGame();  
        }  
        flushGraphics();  
    }  
}
```

Acak.nextInt(19)+1 itu berguna untuk menggenerate angka random, antara 1-20. Lalu fungsi **while({})** itulah yang disebut dengan gameloop, dimana game kita akan terus diulang hingga

kondisi terpenuhi. Maka dari itu, kita menaruh `paintBack()`, `tulisAwal()`, dan `kursorAngka()` di gameloop. Fungsi **`Thread.sleep(GameFrameDelay)`** untuk memberhentikan sejenak gameloop. Dengan begitu, jika `GameFrameDelay` kita isi lebih besar, maka game kita akan lebih lambat pergerakannya, tentu saja penggunaan memory pun akan lebih kecil jika lambat. Tapi tentu saja jangan terlalu lambat, karena akan mengurangi keasyikan bermain nanti. Fungsi `flushGraphics()` akan me-refresh canvas kita, sehingga game kita terlihat bisa bergerak. Jadi sebenarnya urutan dalam gameloop itu,

While(true)

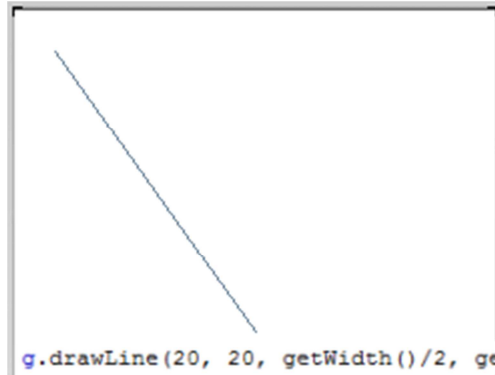
```
{  
    menggambar graphics  
    hapus graphics  
}
```

#Basic Graphics

Untuk membuat gambar di canvas, ada 2 cara, yaitu dengan me-load external file seperti `*.jpg`, `*.png`, dan lain-lain. Dan cara kedua adalah dengan menggunakan fungsi **Graphics** di J2ME. Titik (0,0) pada canvas berada pada kiri atas di layar handphone. Untuk saat ini, kita akan mempelajari tentang bagaimana cara membuat gambar dengan **Graphics**. Ada berbagai bentuk yang dapat kita buat dengan fungsi **Graphics**, yaitu:

1. **`drawLine(int x1, int y1, int x2, int y2)`**

Menggambar garis lurus antara koordinat (x1, y1) dan (x2, y2).



2. **`drawstring(String str, int x, int y, int anchor)`**

Menggambar String di canvas, `str` untuk menulis apa yang kita inginkan pada koordinat (x, y) dan `anchor` itu merupakan posisi dari text kita nanti, berdasarkan koordinat. Untuk sementara, kita pelajari dahulu nilai defaultnya, yaitu 0.

```
g.drawString("J2ME", getWidth()/2, getHeight()/2, 0);
```

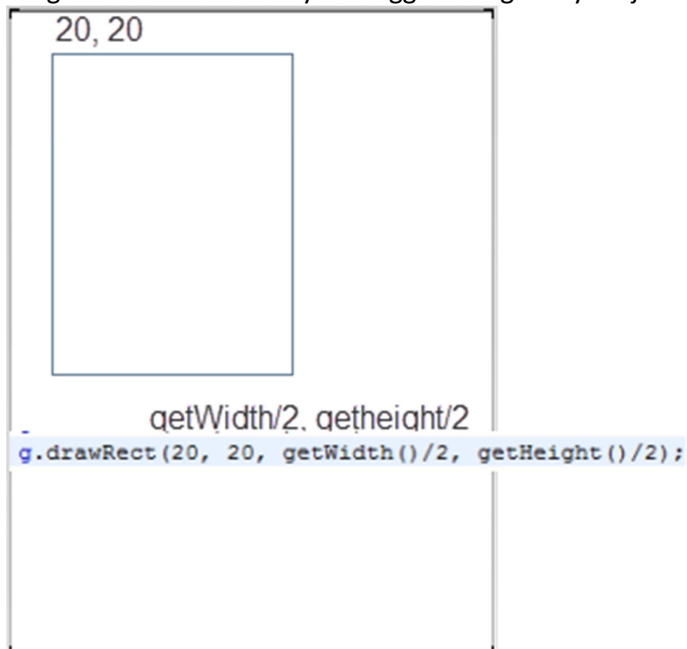


3. **`drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)`** dan **`fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)`**

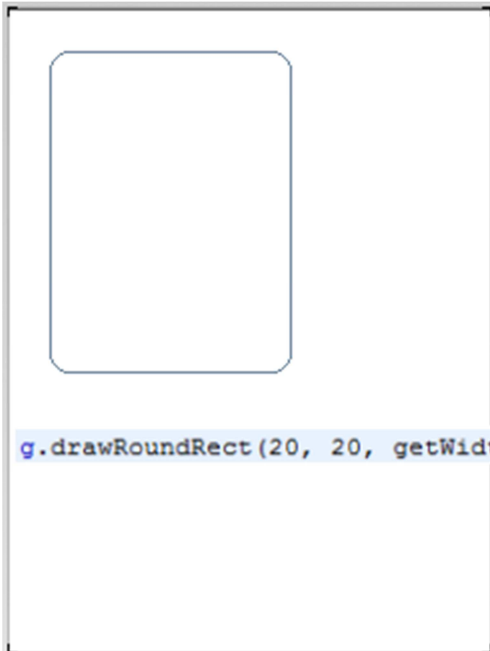
Untuk menggambar lingkaran, pada koordinat (x, y) setinggi **width** selebar **height**, **startAngle** untuk menentukan awal dari penggambaran lingkaran, sedangkan **arcAngle** untuk menentukan berapa derajat lingkaran tersebut akan digambar.



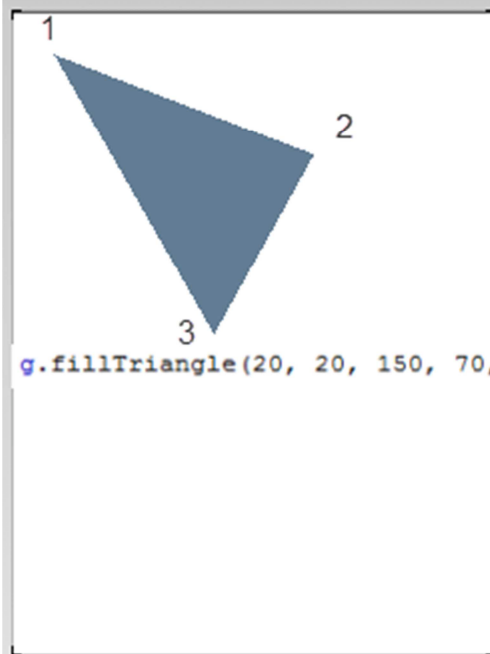
4. **drawRect**(int x, int y, int width, int height) dan **fillRect**(int x, int y, int width, int height)
Pada intinya, fungsi `drawRect` sama dengan fungsi `fillRect`. Tapi yang membedakan ialah pada fungsi `drawRect` kita hanya menggambar garisnya saja. Contohnya seperti ini



5. **drawRoundRect**(int x, int y, int width, int height, int arcWidth, int arcHeight) dan **fillRoundRect**(x, y, width, height, arcWidth, arcHeight)
Menggambar kotak yang memiliki sudut agak membentuk lingkaran. Ada tambahan parameter yaitu `arcWidth` dan `arcHeight`, yang berfungsi untuk menentukan diameter secara horizontal dan vertikal. Contohnya seperti ini.



6. fillTriangle(int x1, int y1, int x2, int y2, int x3, int y3)
Menggambar segitiga dengan titik (x1, y1), (x2, y2), dan (x3, y3)



Tantangan di hari pertama:

Buat permainan tebak angka, aturan:

- Terdiri atas 3 angka yang unik.
- Ketika kita memasukkan jawaban, ada notif bahwa berapa angka yg benar dan berapa angka yg tepat posisi
- Ketika kita menekan atas, maka ada gambar segitiga di atas angkanya.