Agate Mobile Game Developer Camp

Programmer Handout – Day #1

#1 Instalasi Software pendukung:

- Java JDK
- Wireless Toolkit
- Eclipse Pulsar
- Perl
- Nokia S60 Emulator / Emulator default dari Java (Wireless Toolkit)

#2 Add device ke Eclipse:

Dari menu Window > Preferences

Preferences					
type filter text	Device Management				$\Leftrightarrow \bullet \bullet \bullet \bullet \bullet$
General Ant Help Install/Update	Specify the devices to be used by Java ME project Devices not automatically found by Java ME may SDKs:	ts. y be installed by clickin	g the "Manual I	install" buttor	n.
Java Java ME	Name	Default	Configurat	Profile	Edit
Device Management Editors	Manually Installed SDKs	••			Duplicate
New MIDlet Project					Delete
Packaging Preprocessor					Set as Default
Preprocessor Preverification Signing Plug-in Development Run/Debug Tasks Team Usage Data Collector Validation XML					Manual Install
?				ОК	Cancel

Klik Manual Install...

Manual Device Installation Manual Device Installation Select a directory to search for available devices.					
Specify se Devices:	arch directory:				Browse
Install	Group	Name	Configuration	Profile	Select All Deselect All Refresh
?				<u> </u>	Cancel

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

Perduit	configurat	Profile	Edit
			Duplicate
	WMA-2.0	MIDP-2.1	Delete
	WMA-2.0	MIDP-2.1	Set as Default
••			Set as Delaut
			Manual Install
	••	WMA-2.0 WMA-2.0	WMA-2.0 MIDP-2.1 WMA-2.0 MIDP-2.1

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

New MIDle	New MIDlet Project						
Create a M	IIDlet Project						
Enter a proje	ect name.	M					
Project nam	e:						
Applicatio	n Descriptor						
Name to	be used for the jad file, generated during the "Create Package" p	process:					
Use pr	oject name as filename						
🔘 Use cu	istom jad file name						
Jad filena	me:						
.jad							
Contents							
Oreate	new project in workspace						
Create	project from existing source						
Directory	E:\Project Pulsar	Browse					
Configura You can a	tions add more configurations here:						
	SonyEricsson_JP8_240x320_Emu	Add					
		Edit					
		Remove					
Working Sets							
Working Sets:							
?	< Back Next > Finish	Cancel					

#4 Membuat file MIDlet

File	Edit	Source	Refactor	Navigate	Search	Project	Run	Window	Help	
	New				Alt+	Shift+N ▶	M	MIDlet Pro	ject	
	Open	File					C2	Project		
	Close					Ctrl+W	C	Java ME M	IDlet	
	Close	ΔII			Ctrl+9	Shift+W	49	Package		

Membuat file MIDlet bisa dilakukan dengan 2 cara, yaitu:

atau

4 📄	Tebak	Angl	ca	 1		
	Ø site		New	•	M	MIDIet Project
\triangleright	🛋 Jav		Open in New Window		C2	Project
	🖻 Ap		Open Type Hierarchy	F4	Ø	Java ME MIDlet
4 🞥	Tetris		Show In	Alt+Shift+W ►	B	Package
4	🕮 sre	_			C ²	Class

Isi detail dari file MIDlet kita:

👰 New Java ME MI	Dlet	_ D X					
Java ME MIDlet							
Create a New Java ME MIDlet							
Source folder:	Tebak Angka/src	Browse					
Package:	(default)	Browse					
Enclosing type:		Browse					
	1						
Name:							
Modifiers:	public Odefault Oprivate Oprotected						
	abstract final static						
Superclass:	javax.microedition.midlet.MIDlet	Browse					
Interfaces:		Add					
		Remove					
	Which methods should be created?						
	Superclass constructors						
	Unimplemented abstract methods						
	Add To Application Descriptor?						
?	< Back Next > Finish	Cancel					

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 dipause. 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:

📥 Teb	ak An	laka	private st	atic GameCanvas gc;			
⊿ 🚰		New	•	ic boolean running = false;			
4		Go Into		ic int speed = 500;			
		Open in New Window		ead t;			
		Open Type Hierarchy	F4	kAngka() {			
伊		Show In	Alt+Shift+W ►	= Display.getDisplay(this); w GameCanvas(this);			
▷ 🛋		Сору	Ctrl+C	w AboutCanvas();			
e Maria	Ð	Copy Qualified Name		<pre>ommand(new Command("Back", Command.EXI ommandListener(this);</pre>			
010	Ê	Paste	Ctrl+V				
Tut	×	Delete	Delete				
	<u>.</u>	Remove from Context	Ctrl+Alt+Shift+Down	w WelcomeCanvas();			
		Build Path	+	ommand(new Command("Exit", Command.EXI			
		Source	Alt+Shift+S ►	ommand (new Command ("Start", Command.EX			
		Refactor	Alt+Shift+T ►	<pre>.setCurrent(VC);</pre>			
	2	Import		ommandListener(this);			
	4	Export					
	ere.	Refresh	F5	. pauseApp(){ }			
	`	Close Project		destroyApp(boolean b){			
		Assign Working Sets		= false;			
		Due As		tifuDestroyed().			
		Run As	•	Emulated Java ME JAD			
		Debug As	,	2 Emulated Java ME MIDlet			
		Validate		J Java Applet Alt+Shift+X, A			
		Team	*	J 4 Java Application Alt+Shift+X, J			
		Compare With	,	Run Configurations			
		Kestore from Local History					
		WODIIE LOOIS for Java	•	🖉 Tasks 🖳 Console 🐹 🛛 🔗 Search			
		Configure	•	, ume.			
		Properties	Alt+Enter				

#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,

```
privatefinalstaticintPAUSED = 0;
privatefinalstaticintACTIVE = 1;
privatefinalstaticintDESTROYED = 2;
privatestaticintstateTracker = PAUSED;
Command exitCommand = new Command("Exit", Command.EXIT, 2);
private Display gameDisplay;
private Looping gameDoop;
```

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

```
publicvoid startApp(){
    if (gameloop == null)
    {
        gameloop = new Looping(this);
        gameloop.startGame();
        //add some command
        gameloop.addCommand(exitCommand);
```



pauseApp(), untuk memanggil notifyPaused()

```
publicvoid pauseApp()
{
    stateTracker = PAUSED;
    notifyPaused();
}
```

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

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

Looping.java

- Variable yang digunakan yaitu,

```
Graphics g;
privateintkursorMenu=1;
privateintinputDelay = 0;
privateintangka = 0;
Random acak = new Random();
private MIDlet parent;
publiclongGameFrameDelay = 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;
```



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 aka 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).

g.drawLine(20, 20, getWidth()/2, getHeight()/2);

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);
```

J2ME

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.

<pre>g.fillArc(0, 0, getWidth()/2, getWi g.drawArc(getWidth()/2, getHeight()</pre>	dth()/2, 20, 270); /2, getWidth()/2, getWidth()/2, 90, 270);

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.



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.