

Database Perpustakaan

Table1					
isbn	judul	pengarang	penerbit	quantity	tipe
9791644587	12 langkah membangun aplikasi dengan microsoft access 2003	Yuniar Supardi	D@takom	7	komputer
9793767078	12 Proyek Unik Visual Basic 6.0	Firdaus	Maxikom	2	Komputer
9789792711066	150 Rahasia Pemrograman Java	Didik Dwi Prasetyo	Elex Media Komputindo	5	Komputer
9793766760	90 Trik Tersembunyi Photoshop	Chandra	Maxikom	0	Komputer
9785554164	Aplikasi pemrograman internet berbasis PHP	M yusuf	Elekmedia Komputindo	4	komputer
9796485451	Basic Java berorientasi objek programming	Isak Riyanto ST	Penerbit ANDI	4	Komputer
9791314654	Belajar Visual basic	M. Shaludin Rosa	Penerbit ANDI	0	Komputer
9791664455	Dasar Pemrograman Java berorientasi objek	Andi	Elex media Komputindo	4	Komputer
9797313980	Dasar Pemrograman Web Dinamis dengan JSP(Java Server Pages)	Abdul Kadir	Penerbit ANDI	8	Komputer

Table1

isbn	judul	pengarang	penerbit	quantity	tipe
21000039	Fisika Jilid 2 Edisi Ketiga	HALLIDAY & RESNICK	Penerbit Erlangga	3	Teknik
9797313840	Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0	Retna Prasetya, Catur Endi Widodo	Penerbit ANDI	3	Komputer
9793767655	Interior Dapur Dengan AutoCAD & 3ds max	Handi Chandra	Maxikom	1	Komputer
97813300774	Java 2 EE dalam aplikasi Enterprise	Isak Riyanto ST	Penerbit ANDI	3	Komputer
9797312569	Java Handbook Konsep Dasar Pemrograman Java	Patrick Naughton	Penerbit ANDI	7	Komputer
97972789553	Kiat Jitu Menyusun Skripsi Jurusan Informatika	Agnes Maria P	Penerbit ANDI	5	Teknik
97988852234	Mahir dalam 7 hari belajar Microsoft Access 2003	MADCOMS	Penerbit ANDI	2	Komputer
9797620844	Manajemen Strategis	J David Hunger	Penerbit ANDI	7	Ekonomi
9797632256	Manajemen sumber daya manusia	DRS Foustino C G	Penerbit ANDI	3	Ekonomi
9797884456	Marketing Intellegent	Frans M Royan	Penerbit ANDI	5	Ekonomi
97984423321	Marketing Scales	Fandy Tjiptono	Penerbit ANDI	5	Ekonomi
97973100698	Mekanikal	IR Sunarno, M.ENG,	Penerbit	2	Teknik

Table1

isbn	judul	pengarang	penerbit	quantity	tipe
	Elektrikal Lanjutan	PH.D.	ANDI		
97977334555	Membaca Saham	Ali Arifin	Penerbit ANDI	0	Ekonomi
9792584889	Membuat Aplikasi Penjualan dengan PHP & MySQL	Bunafit Nugroho	Ardana Media	5	Komputer
97934445645	Merebut dan mempertahankan pelanggan	DR Tribowo Soedjas SE., MM., SPA	Penerbit ANDI	0	Ekonomi
97973266943	Mesin pemindah bahan	ACH Muhib Zainuri	Penerbit ANDI	1	Teknik
9796447754	Metodologi penelitian bisnis	Murti Sumarni	Penerbit ANDI	0	Ekonomi
9793431215	Motivasion games untuk pelatihan manajemen	Andi Soenarno	Penerbit ANDI	1	Ekonomi
97944366522	Multimedia alat untuk meningkatkan keunggulan bersaing	M Suyanto	Penerbit ANDI	0	Ekonomi
9789970074	Panduan Dasar Visual Basic	M rahman	Elekmedia Komputindo	8	Komputer
97866358542	Panduan lengkap pemrograman J2EE	Widodo Budiharto, S.Si, M.kom.	Penerbit ANDI	5	Komputer
97999329074	Pembuatan Aplikasi game menggunakan Flash	Widodo Budiharto	Penerbit ANDI	3	Komputer
97971989902	Pemrograman	Tri	Elekmedia	6	Komputer

Table1

isbn	judul	pengarang	penerbit	quantity	tipe
	dasar Visual C	Wicaksono	komputindo		
9789792700039	Pemrograman SMS Interaktif Berbasis Java	Mohamad Tri Wicaksono	Elex Media Kompuntindo	3	Komputer
9784456632	Pemrograman web berbasis java	Widodo Budiharto	Penerbit andi	7	Komputer
9793338296	Pengolahan Citra Digital dengan Pendekatan Algoritmik	Rinaldi Munir	Penerbit Informatika	8	Komputer
9797560447	Pengolahan Citra Digital menggunakan Visual Basic	Achmad Basuki, Joshua F Palandi	Graha Ilmu	3	Komputer
97972142231	Perancangan Sistem Operasi	Budi H S	Penerbit ANDI	0	Teknik
97975444465	Perilaku Konsumen	Ristiyanti P	Penerbit ANDI	9	Ekonomi
9796547754	Saving Big Blue	Robert Slater	Penerbit ANDI	0	Ekonomi
97932233655	Sistem informasi strategi untuk keunggulan kompetitif	Djogiyanto	Penerbit ANDI	0	Ekonomi
97913134558	Sistem Operasi	Iwan Binanto DKK	Penerbit ANDI	1	Teknik
97933376334	Strategi Pemasaran	Fandi Tjiptono	Penerbit ANDI	4	Ekonomi
97965511265	Teknik Digital	The Pearson Edu INC	Penerbit ANDI	0	Teknik

Table1

isbn	judul	pengarang	penerbit	quantity	tipe
979411467736010131	Teknik Kontrol Automatik Jilid 1 Edisi Kedua	Katsuhiko Ogata	Penerbit Erlangga	0	Teknik
9797453800	Teori - Aplikasi manajemen proyek konstruksi	Wulfram I Evrianto	Penerbit ANDI	0	Ekonomi
9796554882	Teori dan contoh soal teknik elektro menggunakan matlab	Cekmas Cekdin	Penerbit ANDI	0	Teknik
9797313222	Time Table For Marketing Plan	Frans M Royan	Penerbit ANDI	7	Ekonomi
9789792711158	Tip & Trik Pemrograman Java 2	Didik Dwi Prasetyo	Elex Media Komputindo	3	Komputer
9795334921	Tips dan trik Matlab	Wahyu A P	Penerbit ANDI	0	Teknik
9793767650	Visual Basic 6.0 untuk Orang Awam	Firdaus	Maxikom	4	Komputer

Susunan Program

Script ANT

```
<project name="ta" basedir="." default="compile">

    <path id="lib">
        <fileset dir="${basedir}/lib">
            <include name="*.jar" />
        </fileset>
        <fileset dir="C:\apache-tomcat-6.0.13\lib">
            <include name="*.jar" />
        </fileset>
    </path>

    <target name="compile">
        <javac
            destdir="${basedir}/classes"
            classpathref="lib"
            >
            <src path="${basedir}/src" />
        </javac>
    </target>

</project>
```

Script Web.XML

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-
app_2_4.xsd">

    <display-name>
ta</display-name>

    <servlet>

        <description>
</description>

        <display-name>
Receiver</display-name>

        <servlet-name>Receiver</servlet-name>

        <servlet-class>
ta.servlets.Receiver</servlet-class>

    </servlet>

    <servlet-mapping>

        <servlet-name>Receiver</servlet-name>

        <url-pattern>/Receiver</url-pattern>

    </servlet-mapping>

    <welcome-file-list>

        <welcome-file>index.html</welcome-file>

        <welcome-file>index.htm</welcome-file>

        <welcome-file>index.jsp</welcome-file>

        <welcome-file>default.html</welcome-file>

        <welcome-file>default.htm</welcome-file>

        <welcome-file>default.jsp</welcome-file>
```

</welcome-file-list>

</web-app>

Script Config.Properties

#####

File ini berisi data konfigurasi yg bisa diganti SEBELUM server tomcat

dijalankan.

#####

Connection string ke database, pastikan koneksi ini exist, atau program

akan gagal menghubungi database.

Driver yg di-support hanya sun.jdbc.odbc.JdbcOdbcDriver.

SQLDialect yg di-support hanya Ms Access.

constring=jdbc:odbc:perpus

URL utk ngirim SMS via NowSMS

NowSMSUrl=http://localhost:8800/

Limit pencarian per judul & ISBN, isi HANYA DENGAN ANGKA BULAT.

Limit default adalah 2.

limit=2

Pemisah utk judul buku, tiap judul dipisah oleh karakter

bisa lebih dari 1 karakter, spasi juga bisa di-include. Masing - masing karakter

harus di taruh di dlm kotak [].

separator.judul=[*]

Script Servlet

```
package ta.servlets;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.httpclient.URIException;
import org.apache.commons.httpclient.util.URIUtil;

import ta.Config;
import ta.Utils;
import ta.perpus.Buku;
import ta.perpus.DbBuku;
import ta.perpus.ReqMsg;
import ta.perpus.SendMsg;
import ta.perpus.Sender;

/**
 * Servlet yg bertanggung jawab untuk memproses request pencarian buku.
 * Servlet ini juga bertanggung jawab untuk mengirim hasil pencarian.
 *
 */

public class Receiver extends javax.servlet.http.HttpServlet implements javax.servlet.Servlet {
    static final long serialVersionUID = 1L;
```

```

/**
 * Prefix sms menandakan pencarian berdasarkan kode (isbn).
 */
private static final String SEARCH_KODE = "kode";

/**
 * Prefix sms menandakan pencarian berdasarkan judul buku.
 */
private static final String SEARCH_JUDUL = "buku";

/* (non-Java-doc)
 * @see javax.servlet.http.HttpServlet#HttpServlet()
 */
public Receiver() {
    super();
}

/* (non-Java-doc)
 * @see javax.servlet.http.HttpServlet#doGet(HttpServletRequest request,
HttpServletRequest response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    process(request, response);
}

/* (non-Java-doc)

```

```

        * @see javax.servlet.http.HttpServlet#doPost(HttpServletRequest request,
        HttpServletResponse response)

        */

        protected void doPost(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException

        {

            process(request, response);

        }

        /**

        * Memproses query dari NowSMS.

        */

        private void process(HttpServletRequest req, HttpServletResponse res)

            throws ServletException

        {

            ReqMsg msg = ReqMsg.parse( req );

            System.out.println( "DEBUG: Receive message: " + msg.toString() );

            // hanya proses msg yg valid, jika tidak valid, kirim notifikasi sms tidak valid.

            if ( isValid(msg) )

            {

                System.out.println( "DEBUG: Message valid." );

                int limit;

                try {

                    limit = Integer.parseInt( Config.get("limit") );

                    limit = limit <= 0 ? 2 : limit;

                } catch (NumberFormatException ex) {

                    limit = 2;

                }

            }

        }

```

```

    }

    String sep = "[*]"; // default utk isbn (kode).
    if (msg.getPrefix().equalsIgnoreCase(SEARCH_JUDUL))
        sep = Config.get("separator.judul", "[*]");

    String[] searchs = msg.getIsbn().split( sep );
    for ( String term : searchs )
    {
        /** ada kemungkinan spasi terakhir berisi data kosong, maka
         * bila spasi kosong akan dilewat.
         */
        if (!Utils.isEmpty( term ) && limit > 0) {
            doFind( msg, term, msg.getPrefix() );
            limit--;
        }
    }
}
else
{
    System.out.println( "DEBUG: Message NOT valid." );
    sendGaValid( msg );
}
}

/**
 * Menentukan apakah sms yg dikirim valid atau tidak.
 */

```

```

private boolean isValid(ReqMsg msg)
{
    return
        (!Utils.isEmpty( msg.getSender() )) &&
            (!Utils.isEmpty( msg.getPrefix() )) &&
                (
                    msg.getPrefix().equalsIgnoreCase( SEARCH_KODE ) ||
                    msg.getPrefix().equalsIgnoreCase( SEARCH_JUDUL )
                )
            ) &&
        (!Utils.isEmpty( msg.getIsbn() ));
}

/**
 * Pencarian buku berdasarkan judul dan ISBN, jika ketemu, maka langsung kirim
 * pesan ke sender, begitu juga jika tidak ketemu.
 */
private void doFind(ReqMsg msg, String search, String prefix)
{
    System.out.print( "DEBUG: Searching " + prefix + " " + search + " ..." );

    // Pencarian buku berdasarkan prefix judul || isbn
    String col = "";
    if ( prefix.equalsIgnoreCase(SEARCH_JUDUL) )
        col = "judul";
    else if ( prefix.equalsIgnoreCase(SEARCH_KODE) )
        col = "isbn";
}

```

```

// Bila ketemu = sendDataBuku, jika tidak ketemu = sendGakKetemu
Buku b = DbBuku.search( search, col );
if (b != null) {
    System.out.println( " found." );
    sendDataBuku( msg, b );
} else {
    System.out.println( " NOT found." );
    sendGaKetemu( msg );
}
}

/**
 * Kirim data buku ke requester.
 */
private void sendDataBuku(ReqMsg msg, Buku b)
{
    String status = ( b.getQty() > 0 ? "Tersedia" : "Dipinjam" );

    String s =
        "\"" + b.getNama() + "\", " +
        "\"" + b.getPengarang() + "\", " +
        "\"" + b.getPenerbit() + "\", " +
        "\"" + status + "\", " +
        "\"" + b.getTipe()+ "\"";

    SendMsg send = new SendMsg();

```

```

        send.setDest( msg.getSender() );
        send.setMsg( s );

        Sender.send( send );
    }

    /**
     * Kirim notifikasi, buku tidak ketemu.
     */
    private void sendGaKetemu(ReqMsg msg)
    {
        String s =
            "Maaf, buku yang Anda cari tidak terdapat dalam " +
            "database kami.";

        SendMsg send = new SendMsg();
        send.setDest( msg.getSender() );
        send.setMsg( s );

        Sender.send( send );
    }

    /**
     * Kirim notifikasi format sms yg dikirim tidak valid.
     */
    private void sendGaValid(ReqMsg msg)
    {
        String s =

```

"Format sms yang Anda masukkan salah, Ketik:
Buku<spasi>JUDUL1*JUDUL2 atau Kode<spasi>ISBN1*ISBN2";

```
        SendMsg send = new SendMsg();  
        send.setDest( msg.getSender() );  
        send.setMsg( s );  
  
        Sender.send( send );  
    }  
}
```

Script Config

```
package ta;  
  
import java.io.IOException;  
import java.util.Properties;  
  
/**  
 * Class yang digunakan untuk mengambil nilai dari config file.  
 *  
 */  
public final class Config  
{  
    private static Properties prop;  
  
    private Config() {}  
  
    static
```



```

{
    prop = new Properties();
    try
    {
        prop.load( Config.class.getResourceAsStream("/config.properties") );
    }
    catch (IOException e)
    {
        System.out.println( "WARNING: config.properties tidak ketemu." );
    }
}

/**
 * Untuk mendapatkan nilai utk <code>key</code> dari config file.
 *
 * @return value utk key dari config file, NULL jika tidak ketemu.
 */
public static String get(String key)
{
    String val = prop.getProperty( key );

    if (val == null)
        System.out.println( "WARNING: property " + key + " tidak ketemu." );

    return val;
}

/**

```

```

    * Digunakan untuk nilai utk <code>key</code> dari config file, dan akan
    * menggunakan <code>def</code> jika tidak ketemu atau nilainya kosong.
    *
    * @return value utk key dari config file, NULL jika tidak ketemu.
    */
public static String get(String key, String def)
{
    String val = Config.get(key);
    if (val == null || val.trim().length() <= 0)
        val = def;

    return val;
}
}

```

Script ConMgr

```

package ta;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 * Connection manager, menyediakan connection pool to db.
 *
 */

```

```

public final class ConMgr
{
    private ConMgr() {}

    static
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        }
        catch (ClassNotFoundException e)
        {
            System.out.println("Koneksi Salah");
        }
    }

    /**
     * Membuka koneksi ke database, setting connection url dapat dilakukan
     * di config file. koneksi yg sudah selesai dipakai harus ditutup sendiri
     * @return koneksi ke db yg sudah siap pakai.
     * @throws RuntimeException jika tidak mendapatkan koneksi atau jika ada
     * SQLException.
     */
    public static Connection getCon()
        throws RuntimeException
    {
        String constr = Config.get( "constring" );

        try

```

```

        {
            Connection con = DriverManager.getConnection( constr );
            if (con == null)
                throw new RuntimeException("ERROR: Tidak dapat koneksi
ke " + constr);

            return con;
        }
        catch (SQLException e)
        {
            e.printStackTrace();
            throw new RuntimeException(e);
        }
    }
}

```

Script Class Utils

```

package ta;

/**
 * Util class.
 *
 */
public final class Utils
{
    private Utils() {}
}

```

```

/**
 * Check apakah String kosong ato tidak.
 *
 * @param str
 * @return TRUE kalo NULL ato zero-length String(String<=0).
 */
public static boolean isEmpty(String str)
{
    return (str == null || str.trim().length() <= 0);
}
}

```

Script Class – Class yang terdapat dalam perpus

Script Class Buku

```

package ta.perpus;

import java.io.Serializable;

/**
 * Wraps table buku di db.
 *
 */
public class Buku implements Serializable
{

```

```
private String isbn;  
private String nama;  
private String pengarang;  
private String penerbit;  
private int qty;  
private String tipe;
```

```
public Buku() { }
```

```
/**
```

```
 * No ISBN buku.
```

```
 *
```

```
 * @return
```

```
 */
```

```
public String getIsbn()
```

```
{
```

```
    return isbn;
```

```
}
```

```
public void setIsbn(String isbn)
```

```
{
```

```
    this.isbn = isbn;
```

```
}
```

```
/**
```

```
 * Nama buku.
```

```
 *
```

```
 * @return
```

```

*/

public String getNama()
{
    return nama;
}

public void setNama(String nama)
{
    this.nama = nama;
}

/**
 * Pengarang buku.
 *
 * @return
 */

public String getPengarang()
{
    return pengarang;
}

public void setPengarang(String pengarang)
{
    this.pengarang = pengarang;
}

/**
 * Penerbit buku.

```

```

*
* @return
*/
public String getPenerbit()
{
    return penerbit;
}

public void setPenerbit(String penerbit)
{
    this.penerbit = penerbit;
}

/**
 * Qty (ketersediaan) buku.
 *
 * @return
 */
public int getQty()
{
    return qty;
}

public void setQty(int qty)
{
    this.qty = qty;
}

```



```

/**
 * Tipe buku (komik, pelajaran, dsb).
 *
 * @return -1 jika data di db kosong.
 */
public String getTipe()
{
    return tipe;
}

public void setTipe(String tipe)
{
    this.tipe = tipe;
}
}

```

Script DbBuku

```

package ta.perpus;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import ta.ConMgr;

/**

```

```

* Berisi operasi2 interaksi ke database khusus table buku.
*/

public final class DbBuku
{
    private DbBuku() {}

    /**
     * Digunakan untuk mencari buku berdasarkan kolom <code>col</code> yg berisi nilai
     * <code>search</code>.
     * @param search nilai parameter pencarian, case INsensitive.
     * @param col kolom yg hendak di cari, NOTE: kolom ini harus ada pada table Buku.
     *
     * @return NULL jika tidak ketemu.
     */
    public static Buku search(String search, String col)
    {
        Connection con = ConMgr.getCon();

        String q = "SELECT * FROM Table1 AS t WHERE LCase(t." + col + ") = ?";

        try
        {
            PreparedStatement ps = con.prepareStatement(q);
            ps.setString( 1, search.toLowerCase().trim() );

            ResultSet rs = ps.executeQuery();
            if (rs.next())
                return convert( rs );
            else

```

```

        return null;
    }
    catch (SQLException e)
    {
        e.printStackTrace();
        throw new RuntimeException(e);
    }
    finally {
        try {
            if (con != null)
                con.close();
        } catch (SQLException e) {
            // Error SQL Statement.
        }
    }
}

```

```

/**

```

```

 * Konvert bentuk ResultSet ke bentuk Buku.

```

```

 *

```

```

 * @param rs

```

```

 * @return object buku yg siap pakai.

```

```

 */

```

```

private static Buku convert(ResultSet rs)

```

```

    throws SQLException

```

```

{

```

```

    Buku b = new Buku();

```

```

        b.setIsbn( rs.getString("isbn" ) );
        b.setNama( rs.getString("judul" ) );
        b.setPenerbit( rs.getString("penerbit" ) );
        b.setPengarang( rs.getString("pengarang" ) );
        b.setQty( rs.getInt("quantity" ) );

        b.setTipe( rs.getString("tipe" ) );

        return b;
    }
}

```

Script Class ReqMsg

```

package ta.perpus;

import java.io.Serializable;

import javax.servlet.http.HttpServletRequest;

/**
 * ReqMsg adalah kelas wrapper utk query yg diterima dari
 * NowSMS.
 *
 */
public class ReqMsg implements Serializable
{

```

```

private String sender;

private String prefix;

private String isbn;

private ReqMsg() {}

/**
 * Proses sms dari NowSMS query.
 *
 * @param req request dari NowSMS.
 *
 * @return
 */
public static ReqMsg parse(HttpServletRequest req)
{
    ReqMsg msg = new ReqMsg();

    msg.setSender( req.getParameter("sender") );
    msg.setPrefix( req.getParameter("prefix") );
    msg.setIsbn( req.getParameter("sms") );

    return msg;
}

/**
 * No. HP pengirim request.
 *
 * @return NULL kalau kosong

```

```

*/
public String getSender()
{
    return sender;
}

public void setSender(String sender)
{
    this.sender = sender;
}

/**
 * Perintah pengiriman.
 *
 * @return NULL jika kosong.
 */
public String getPrefix()
{
    return prefix;
}

public void setPrefix(String prefix)
{
    this.prefix = prefix;
}

/**
 * No. ISBN atau Judul Buku yg hendak dicari.

```

```

    *
    * @return NULL jika kosong.
    */
    public String getIsbn()
    {
        return isbn;
    }

    public void setIsbn(String isbn)
    {
        this.isbn = isbn;
    }

    public String toString()
    {
        return sender + ", " + prefix + ", " + isbn;
    }
}

```

Script Class Sender

```

package ta.perpus;

import java.io.IOException;

import ta.Config;

import org.apache.commons.httpclient.HttpClient;

```

```

import org.apache.commons.httpclient.HttpException;

import org.apache.commons.httpclient.HttpMethod;

import org.apache.commons.httpclient.NameValuePair;

import org.apache.commons.httpclient.methods.GetMethod;

/**
 * Kelas yg menangani pengiriman pesan ke NowSMS.
 *
 */

public final class Sender
{
    private Sender() {}

    /**
     * Kirim pesan ke NowSMS.
     *
     * @param msg
     */

    public static void send(SendMsg msg)
    {
        System.out.println( "DEBUG: Sending reply to " + msg.getDest() +
            ", content: " + msg.getMsg() );

        HttpMethod method = new GetMethod( Config.get("NowSMSUrl") );

        NameValuePair[] qs = new NameValuePair[2];
        qs[0] = new NameValuePair( "PhoneNumber", msg.getDest() );
        qs[1] = new NameValuePair( "Text", msg.getMsg() );
    }
}

```



```
        method.setQueryString(qs);
        HttpClient client = new HttpClient();
        try
        {
            client.executeMethod(method);
        }
        catch (HttpException e)
        {
            e.printStackTrace();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

Script Class SendMsg

```
package ta.perpus;

import java.io.Serializable;

public class SendMsg implements Serializable
{
    private String dest;
```

```

private String msg;

public SendMsg() {}

/**
 * Nomor tujuan.
 *
 * @return
 */
public String getDest()
{
    return dest;
}

public void setDest(String dest)
{
    this.daest = dest;
}

/**
 * Pesan yg hendak dikirim.
 * @return
 */
public String getMsg()
{
    return msg;
}

```

```
public void setMsg(String msg)
{
    this.msg = msg;
}
}
```