

DAFTAR PUSTAKA

Andreas Handoyo, Gregorius Satia Budhi, Hendra Rusly. Aplikasi *Data Mining* untuk meneliti Asosiasi Pembelian Item Barang di Supermarket dengan Metode *Market Basket Analysis*.

Budi Dwi Satoto, Daniel O Siahaan, Akhmad Saikhu. 2010. Perbaikan Struktur *Weighted Tree* dengan Metode Partisi Fuzzy dalam Pembangkitan *Frequent Itemset*. Jurnal Ilmiah Cursor Menuju Solusi Teknologi Informasi : Vol. 5, No. 3.

Emha Taufiq Luthfi. 2009. Penerapan Data Mining Algoritma Asosiasi Untuk Meningkatkan Penjualan. Jurnal Dasi : Vol. 10, No. 1

Emha Taufiq Luthfi, Kusri. 2009. Algoritma Data Mining. Andi Yogyakarta.

Firdaus Pratama, Afriyudi, Evi Yulianingsih. Data Mining dengan menerapkan Hash Based pada Persediaan Barang di Apotik Srikandi.

Gregorius Satia Budhi, Andreas Handoko, Cristine Oktavina Wirawan. 2009. Algoritma *Generalized Sequential Pattern* untuk menggali Data Sekuensial Sirkulasi Buku pada Perpustakaan UK Petra. Yogyakarta : Seminar Nasional Aplikasi Teknologi Informasi.

Ibnu Rachman Chalid. 2009. Aplikasi Audio Steganografi untuk melindungi Data menggunakan Bahasa Pemrograman Java. http://www.gunadarma.ac.id/library/articles/graduate/industrial-technology/2009/Artikel_50405368.pdf (diakses pada tgl 4 Januari 2012 pukul 09.00 WITA)

Larose, Daniel T. 2005. *Discovering Knowledge in Data : An Introduction to Data Mining*. John Willey & Sons, Inc

Tinambunan, Wesley N. 2010. Model Persediaan Dengan Backorder Berdasarkan Defuzzifikasi Signed Distance Method. <http://repository.usu.ac.id/handle/123456789/18975>. (diakses tgl 2 Januari 2012 pukul 14.00 WITA)

Lampiran 1
Tabel data barang

kode	nama	harga
00014	Large Rice Bowl with Spoon	179000
00024	New Shelf Saver (2)	115000
00030	Small Square Round (4)	150000
00031	Trio Rio	110000
00032	Smart Severs (4)	100000
00036	Medium Stak N Stor (2)	199000
00038	Large Stak N Stor	115000
00039	Jumbo Stak N Stor	138000
00040	New Snak Stor	155000
00071	Modular Mates Square 1	90000
00072	Modular Mates Square 2	109000
00073	Modular Mates Square 3	137000
00074	Modular Mates Oval 1	57000
00075	Modular Mates Oval 2	72000
00076	Modular Mates Oval 3	89000

Lampiran 2

Tabel data barang *sales force*

kode	tersedia
00014	0
00024	0
00030	0
00031	0
00032	0
00036	0
00038	0
00039	0
00040	1
00071	0
00072	0
00073	0
00074	0
00075	0

Lampiran 3
Tabel data barang distributor

kode	tersedia
00014	1
00024	1
00030	1
00031	0
00032	1
00036	1
00038	1
00039	1
00040	1
00071	1
00072	1
00073	1
00074	0
00075	0

Lampiran 4

Tabel histori penjualan

tanggal	kodebrg
2012-12-10	00102
2012-12-10	00079
2012-12-10	00266
2012-12-10	00077
2012-12-17	00079
2012-12-17	00073
2012-12-20	00239
2012-12-20	00079
2012-12-22	00256
2012-12-21	00162
2012-12-21	00263
2012-12-17	00286
2012-12-26	00165

Lampiran 5

Tabel data set

transaksi	isi
1	00014
2	0002400073
3	0003000239
4	000300025900289
5	00030
6	0003000239
7	00280
8	000300010500162
9	0001400273
10	0002400105001620016400165
11	002390027100272
12	002390026600288
13	00250
14	000300027100272
15	00024

Lampiran 6

Tabel C1L1

itemset	jumlah
00014	14
00024	9
00030	15
00036	3
00038	3
00072	3
00073	7
00081	4
00105	8
00162	8
00164	6
00165	6
00213	8
00217	10

Lampiran 7

Tabel C2L2

itemset	jumlah
0010500162	7
0016400165	6
0001400249	4
0001400265	4
0002400217	4
0027600277	3
0027100272	3
0010500165	3
0010500164	3
0007200073	3
0003000239	3
0028000288	3

Lampiran 8

Sourcecode MenuBarang.Java

Nama Activity	Modul
Data Barang	MenuBarang.Java
<pre> package wana.uke; import java.awt.EventQueue; import javax.swing.JFrame; import javax.swing.JPanel; import javax.swing.border.EmptyBorder; import javax.swing.table.DefaultTableModel; import javax.swing.JLabel; import javax.swing.JOptionPane; import javax.swing.JTextField; import javax.swing.JScrollPane; import javax.swing.JTable; import javax.swing.JButton; import javax.swing.UIManager; import java.awt.event.ActionListener; import java.awt.event.ActionEvent; import java.sql.*; import java.awt.event.MouseAdapter; import java.awt.event.MouseEvent; @SuppressWarnings("serial") public class MenuBarang extends JFrame { private JPanel contentPane; private JTextField tfKode; private JTextField tfNama; private JTextField tfHarga; private JTable tabel; //buat header String header[] = {"Kode", "Nama Barang", "Harga Satuan"}; DefaultTableModel tabelModel; private JButton btnUpdate; private JButton btnHapus; /** * Launch the application. */ public static void main(String[] args) { EventQueue.invokeLater(new Runnable() { public void run() { try { UIManager.setLookAndFeel("com.jtattoo.plaf.mcwin.McWinLookAndFeel </pre>	

```

");
        MenuBarang frame = new MenuBarang();
        frame.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
    });
}

/**
 * Create the frame.
 */
public MenuBarang() {
    setResizable(false);
    setTitle("Data Barang");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 592, 536);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblKodeBarang = new JLabel("Kode Barang");
    lblKodeBarang.setBounds(10, 11, 83, 14);
    contentPane.add(lblKodeBarang);

    tfKode = new JTextField();
    tfKode.setBounds(101, 8, 60, 20);
    contentPane.add(tfKode);
    tfKode.setColumns(10);

    JLabel lblNamaBarang = new JLabel("Nama Barang");
    lblNamaBarang.setBounds(10, 39, 83, 14);
    contentPane.add(lblNamaBarang);

    tfNama = new JTextField();
    tfNama.setColumns(10);
    tfNama.setBounds(101, 36, 246, 20);
    contentPane.add(tfNama);

    JLabel lblHargaBarang = new JLabel("Harga Barang");
    lblHargaBarang.setBounds(10, 67, 83, 14);
    contentPane.add(lblHargaBarang);

    tfHarga = new JTextField();
    tfHarga.setColumns(10);
    tfHarga.setBounds(101, 64, 120, 20);
    contentPane.add(tfHarga);

    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setBounds(10, 92, 556, 361);
    contentPane.add(scrollPane);

```

```

//tampilkan header pada tabel
tabelModel = new DefaultTableModel(null, header);
tabel = new.JTable();
tabel.setModel(tabelModel);
scrollPane.setViewportView(tabel);

tabel.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        getData();
    }
});

JButton btnSimpan = new JButton("Simpan");
btnSimpan.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        try {
            //simpan data ke database
            Connection konek =
Koneksi.getKoneksi();
            String query = "INSERT INTO barang
VALUES(?,?,?)";
            PreparedStatement prepare =
konek.prepareStatement(query);
            prepare.setString(1,tfKode.getText());
            prepare.setString(2,tfNama.getText());

            prepare.setFloat(3,Float.parseFloat(tfHarga.getText()));
            prepare.executeUpdate();

            String querysf = "INSERT INTO barangsf
VALUES(?,?)";
            PreparedStatement preparesf =
konek.prepareStatement(querysf);
            preparesf.setString(1,tfKode.getText());
            preparesf.setInt(2,0);
            preparesf.executeUpdate();

            String querydistor = "INSERT INTO
barangdistor VALUES(?,?)";
            PreparedStatement preparedistor =
konek.prepareStatement(querydistor);
            preparedistor.setString(1,tfKode.getText());
            preparedistor.setInt(2,1);
            preparedistor.executeUpdate();

            //JOptionPane.showMessageDialog(null,
            "Data berhasil ditambahkan ke database");
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null,
            "Data gagal ditambahkan ke database");
        }
    }
});

```

```

        ex.printStackTrace();
    }
    finally
    {
        getDataTable();
    }
}
});
btnSimpan.setBounds(10, 464, 89, 23);
contentPane.add(btnSimpan);

btnUpdate = new JButton("Update");
btnUpdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        try {
            //update data
            Connection konek =
Koneksi.getKoneksi();
            String query = "UPDATE barang SET
nama='"+tfNama.getText()+"',
harga="+Float.parseFloat(tfHarga.getText())+" WHERE kode
='"+tfKode.getText()+"'";
            PreparedStatement prepare =
konek.prepareStatement(query);

            prepare.executeUpdate();
            JOptionPane.showMessageDialog(null,
>Data berhasil diupdate");

            prepare.close();
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null,
>Data gagal diupdate");
            ex.printStackTrace();
        }
    }
    finally
    {
        getDataTable();
    }
}
});
btnUpdate.setBounds(132, 464, 89, 23);
contentPane.add(btnUpdate);

btnHapus = new JButton("Hapus");
btnHapus.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            //hapus data
            Connection konek =
Koneksi.getKoneksi();
            String query = "DELETE FROM barang
WHERE kode ="+tfKode.getText();
            PreparedStatement prepare =

```

```

konek.prepareStatement(query);
        prepare.executeUpdate();
        prepare.close();

        String queriesf = "DELETE FROM barangsf
WHERE kode =" + tfKode.getText();
        PreparedStatement preparesf =
konek.prepareStatement(queriesf);
        preparesf.executeUpdate();
        preparesf.close();

        String querydistor = "DELETE FROM
barangdistor WHERE kode =" + tfKode.getText();
        PreparedStatement preparedistor =
konek.prepareStatement(querydistor);
        preparedistor.executeUpdate();
        preparedistor.close();

        JOptionPane.showMessageDialog(null,
>Data berhasil dihapus");
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null,
>Data gagal dihapus");
        ex.printStackTrace();
    }
    finally
    {
        getDataTable();
    }
    });
    btnHapus.setBounds(258, 464, 89, 23);
    contentPane.add(btnHapus);
    getDataTable();
} //akhir konstruktur

//isi tabel
public void getDataTable()
{
    tabelModel.getDataVector().removeAllElements();
    tabelModel.fireTableDataChanged();
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();
        String query = "SELECT * FROM barang";
        ResultSet rs = state.executeQuery(query);
        while(rs.next())
        {
            Object obj[] = new Object[3];
            obj[0] = rs.getString(1);
            obj[1] = rs.getString(2);
            obj[2] = rs.getFloat(3);
        }
    }
}

```

```

        tabelModel.addRow(obj);
    }
    rs.close();
    state.close();
}
catch(Exception e)
{
    e.printStackTrace();
}
}

//isi texfield sesuai tabel
public void getData()
{
    int pilih = tabel.getSelectedRow();
    if(pilih == -1)
    {
        return;
    }
    String kode = (String) tabelModel.getValueAt(pilih, 0);
    tfKode.setText(kode);
    String nama = (String) tabelModel.getValueAt(pilih, 1);
    tfNama.setText(nama);
    Float harga = (Float) tabelModel.getValueAt(pilih, 2);
    tfHarga.setText("" + harga);
}
}
}

```

Lampiran 9

Sourcecode MenuHisjual.Java

Nama Activity	Modul
Data Histori Penjualan	MenuHisjual.Java
<pre> package wana.uke; import java.awt.EventQueue; import javax.swing.JFrame; import javax.swing.JPanel; import javax.swing.border.EmptyBorder; import javax.swing.table.DefaultTableModel; import javax.swing.JLabel; import javax.swing.JOptionPane; import javax.swing.JScrollPane; import javax.swing.JTable; import javax.swing.JButton; import javax.swing.UIManager; import java.awt.event.ActionListener; import java.awt.event.ActionEvent; import java.sql.*; import java.awt.event.MouseAdapter; import java.awt.event.MouseEvent; import de.wannawork.jcalendar.JCalendarComboBox; import javax.swing.JComboBox; @SuppressWarnings("serial") public class MenuHisjual extends JFrame { private JPanel contentPane; private JTable tabel; //buat header String header[] = {"Tanggal", "Kode Barang", "Nama Barang"}; DefaultTableModel tabelModel; private JButton btnHapus; private JCalendarComboBox cbKelender = new JCalendarComboBox(); private JComboBox cbNamabrg = new JComboBox(); String datestr = new String(); /** * Launch the application. */ public static void main(String[] args) { EventQueue.invokeLater(new Runnable() { public void run() { try { UIManager.setLookAndFeel("com.jtattoo.plaf.mcwin.McWinLookAndFeel"); MenuHisjual frame = new MenuHisjual(); frame.setVisible(true); } catch (Exception e) { e.printStackTrace(); } } }); } } </pre>	


```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
}

/**
 * Create the frame.
 */
public MenuHisjual() {
    setResizable(false);
    setTitle("Data Histori Penjualan");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 592, 536);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblKTanggal = new JLabel("Tanggal");
    lblKTanggal.setBounds(10, 11, 83, 14);
    contentPane.add(lblKTanggal);

    JLabel lblNamaBarang = new JLabel("Nama Barang");
    lblNamaBarang.setBounds(10, 39, 83, 14);
    contentPane.add(lblNamaBarang);

    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setBounds(10, 64, 556, 389);
    contentPane.add(scrollPane);

    //tampilkan header pada tabel
    tabelModel = new DefaultTableModel(null, header);
    tabel = new JTable();
    tabel.setModel(tabelModel);
    scrollPane.setViewportViewView(tabel);

    tabel.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
            getData();
        }
    });

    JButton btnSimpan = new JButton("Simpan");
    btnSimpan.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            try {
                //simpan data ke database

                Connection konek =
Koneksi.getKoneksi();
                Statement state =

```

```

konek.createStatement();
String querytmp = "SELECT kode FROM
barang WHERE nama='"+cbNamabrg.getSelectedItem().toString()+"'";
ResultSet rs =
state.executeQuery(querytmp);
String kodetmp="";
while(rs.next())
{
    kodetmp = rs.getString("kode");
}
rs.close();
state.close();
int tgl =
cbKelender.getDate().getDate();
int bln =
cbKelender.getDate().getMonth();
int thn =
cbKelender.getDate().getYear();
java.sql.Date date = new
java.sql.Date(thn, bln, tgl);
String query = "INSERT INTO hisjual
VALUES(?,?)";
PreparedStatement prepare =
prepare.setDate(1, date);
prepare.setString(2, kodetmp);
prepare.executeUpdate();

//JOptionPane.showMessageDialog(null,
>Data berhasil ditambahkan ke database");
} catch (Exception ex) {
    JOptionPane.showMessageDialog(null,
>Data gagal ditambahkan ke database");
    System.out.println(datestr);
    ex.printStackTrace();
}
finally
{
    getDataTable();
}
});
btnSimpan.setBounds(10, 464, 89, 23);
contentPane.add(btnSimpan);

btnHapus = new JButton("Hapus");
btnHapus.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            //ambil kode
            Connection konek =
Koneksi.getKoneksi();
Statement state =

```

```

konek.createStatement();
String querytmp = "SELECT kode
FROM barang WHERE nama='"+cbNamabrg.getSelectedItem().toString()+"'";
ResultSet rs =
state.executeQuery(querytmp);
String kodetmp="";
while(rs.next())
{
    kodetmp =
rs.getString("kode");
}
rs.close();
state.close();

//ambil tanggal
int tgl =
cbKelender.getDate().getDate();
int bln =
cbKelender.getDate().getMonth();
int thn =
cbKelender.getDate().getYear();
java.sql.Date date = new
String datestr = new

//hapus data
String query = "DELETE FROM
hisjual WHERE kodebrg='"+kodetmp+"' AND tanggal='"+datestr+"'";
PreparedStatement prepare =
prepare.executeUpdate();
prepare.close();

JOptionPane.showMessageDialog(null, "Data berhasil dihapus");
} catch (Exception ex) {
JOptionPane.showMessageDialog(null, "Data gagal dihapus");
ex.printStackTrace();
}
finally
{
    getDataTable();
}
});
btnHapus.setBounds(109, 464, 89, 23);
contentPane.add(btnHapus);

cbKelender.setBounds(103, 5, 118, 20);
contentPane.add(cbKelender);

cbNamabrg.setBounds(103, 36, 266, 20);

```

```

        contentPane.add(cbNamabrg);
        getDatatocombo();
        getDataTable();
    }//akhir konstuktur

    //isi tabel
    public void getDataTable()
    {
        tabelModel.getDataVector().removeAllElements();
        tabelModel.fireTableDataChanged();
        try
        {
            Connection konek = Koneksi.getKoneksi();
            Statement state = konek.createStatement();
            String query = "SELECT hisjual.tanggal,
hisjual.kodebrg, barang.nama FROM hisjual INNER JOIN barang ON
hisjual.kodebrg = barang.kode ORDER BY hisjual.tanggal,
hisjual.kodebrg";
            ResultSet rs = state.executeQuery(query);
            while(rs.next())
            {
                Object obj[] = new Object[3];
                obj[0] = rs.getDate(1);
                obj[1] = rs.getString(2);
                obj[2] = rs.getString(3);
                tabelModel.addRow(obj);
            }
            rs.close();
            state.close();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }

    //isi combobox dengan nama barang pada tabel barang
    public void getDatatocombo()
    {
        cbNamabrg.removeAllItems();
        try {
            //ambil data dari database
            Connection konek = Koneksi.getKoneksi();
            Statement state = konek.createStatement();
            String query = "SELECT nama FROM barang";

            ResultSet rs = state.executeQuery(query);
            while(rs.next())
            {
                cbNamabrg.addItem(rs.getString("nama"));
            }
            rs.close();
            state.close();
        } catch (Exception ex) {

```

```
        ex.printStackTrace();
    }
}

//isi combo sesuai tabel yang dipilih
public void getData()
{
    int pilih = tabel.getSelectedRow();
    if(pilih == -1)
    {
        return;
    }
    java.sql.Date date = (java.sql.Date)
tabelModel.getValueAt(pilih, 0);
    cbKelender.setDate(date);
    String namaStr = (String) tabelModel.getValueAt(pilih, 2);
    cbNamabrg.setSelectedItem(namaStr);
}
}
```

Lampiran 10
Sourcecode MenuBarangsف.Java

Nama Activity	Modul
Barang sales force	MenuBarangsف.Java
<pre> package wana.uke; import java.awt.EventQueue; import java.awt.event.MouseAdapter; import java.awt.event.MouseEvent; import java.sql.*; import javax.swing.JFrame; import javax.swing.UIManager; //import javax.swing.JOptionPane; import javax.swing.JPanel; import javax.swing.border.EmptyBorder; import javax.swing.table.DefaultTableModel; import javax.swing.JScrollPane; @SuppressWarnings("serial") public class MenuBarangsف extends JFrame { private JPanel contentPane; private ModifyTable tabel; //buat header String header[] = {"Kode","Nama Barang","Tersedia"}; DefaultTableModel tabelModel; /** * Launch the application. */ public static void main(String[] args) { EventQueue.invokeLater(new Runnable() { public void run() { try { UIManager.setLookAndFeel("com.jtattoo.plaf.mcwin.McWinLookAndFeel"); MenuBarangsف frame = new MenuBarangsف(); frame.setVisible(true); } catch (Exception ex) { ex.printStackTrace(); } } }); } } </pre>	

```

* Create the frame.
*/
public MenuBarangsf() {
    setResizable(false);
    setTitle("Data Ketersediaan Barang Sales Force");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 592, 536);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setBounds(10, 11, 556, 476);
    contentPane.add(scrollPane);

    //tampilkan header pada tabel
    tabelModel = new DefaultTableModel(null, header);
    tabel = new ModifyTable();
    tabel.setModel(tabelModel);
    scrollPane.setViewportView(tabel);
    getDataTable();
    tabel.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
            UpdateData();
        }
    });
} //akhir konstruktur

//isi tabel
public void getDataTable()
{
    tabelModel.getDataVector().removeAllElements();
    tabelModel.fireTableDataChanged();
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();
        String query = "SELECT barangsf.kode, barang.nama,
barangsf.tersedia FROM barangsf INNER JOIN barang ON barangsf.kode =
barang.kode";
        ResultSet rs = state.executeQuery(query);
        while(rs.next())
        {
            Object obj[] = new Object[3];
            obj[0] = rs.getString(1);
            obj[1] = rs.getString(2);
            obj[2] = rs.getBoolean(3);
            tabelModel.addRow(obj);
        }
        rs.close();
        state.close();
    }
}

```

```

        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
    //isi update data setiap mengklik centang
    public void UpdateData()
    {
        int pilih = tabel.getSelectedRow();
        if(pilih == -1)
        {
            return;
        }

        String kode = (String) tabelModel.getValueAt(pilih, 0);
        Object tersedia = tabelModel.getValueAt(pilih, 2);
        int sedia;
        if(tersedia.equals(false)){
            sedia=0;
        }
        else{
            sedia=1;
        }

        try {
            //update data
            Connection konek = Koneksi.getKoneksi();
            String query = "UPDATE barangsf SET
tersedia="+sedia+" WHERE kode='"+kode+"'";
            PreparedStatement prepare =
konek.prepareStatement(query);
            //
            prepare.setString(1,kode);
            //
            prepare.setInt(2, sedia);
            prepare.executeUpdate();
            //
            JOptionPane.showMessageDialog(null, "Data berhasil
diupdate");
            prepare.close();
        } catch (Exception ex) {
            ex.printStackTrace();
            //
            JOptionPane.showMessageDialog(null, "Data gagal
diupdate");
        }
    }
}

```


Lampiran 11

Sourcecode MenuBarangdistor.Java

Nama Activity	Modul
Barang distributor	MenuBarangdistor.Java
<pre> package wana.uke; import java.awt.EventQueue; import java.awt.event.MouseAdapter; import java.awt.event.MouseEvent; import java.sql.*; import javax.swing.JFrame; //import javax.swing.JOptionPane; import javax.swing.JPanel; import javax.swing.UIManager; import javax.swing.border.EmptyBorder; import javax.swing.table.DefaultTableModel; import javax.swing.JScrollPane; @SuppressWarnings("serial") public class MenuBarangdistor extends JFrame { private JPanel contentPane; private ModifyTable tabel; //buat header String header[] = {"Kode", "Nama Barang", "Tersedia"}; DefaultTableModel tabelModel; /** * Launch the application. */ public static void main(String[] args) { EventQueue.invokeLater(new Runnable() { public void run() { try { UIManager.setLookAndFeel("com.jtattoo.plaf.mcwin.McWinLookAndFeel "); MenuBarangdistor frame = new MenuBarangdistor(); frame.setVisible(true); } catch (Exception ex) { ex.printStackTrace(); } } }); } </pre>	

```

    }

    /**
     * Create the frame.
     */
    public MenuBarangdistor() {
        setResizable(false);
        setTitle("Data Ketersediaan Barang Distributor");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 592, 536);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(10, 11, 556, 476);
        contentPane.add(scrollPane);

        //tampilkan header pada tabel
        tabelModel = new DefaultTableModel(null, header);
        tabel = new ModifyTable();
        tabel.setModel(tabelModel);
        scrollPane.setViewportView(tabel);
        getDataTable();
        tabel.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent arg0) {
                UpdateData();
            }
        });
    } //akhir konstruktur

    //isi tabel
    public void getDataTable()
    {
        tabelModel.getDataVector().removeAllElements();
        tabelModel.fireTableDataChanged();
        try
        {
            Connection konek = Koneksi.getKoneksi();
            Statement state = konek.createStatement();
            String query = "SELECT barangdistor.kode,
barang.nama, barangdistor.tersedia FROM barangdistor INNER JOIN barang
ON barangdistor.kode = barang.kode";
            ResultSet rs = state.executeQuery(query);
            while(rs.next())
            {
                Object obj[] = new Object[3];
                obj[0] = rs.getString(1);
                obj[1] = rs.getString(2);
                obj[2] = rs.getBoolean(3);
                tabelModel.addRow(obj);
            }
        }
    }

```

```

        rs.close();
        state.close();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
//isi update data setiap mengklik centang
public void UpdateData()
{
    int pilih = tabel.getSelectedRow();
    if(pilih == -1)
    {
        return;
    }

    String kode = (String) tabelModel.getValueAt(pilih, 0);
    Object tersedia = tabelModel.getValueAt(pilih, 2);
    int sedia;
    if(tersedia.equals(false)){
        sedia=0;
    }
    else{
        sedia=1;
    }

    try {
        //update data
        Connection konek = Koneksi.getKoneksi();
        String query = "UPDATE barangdistor SET
tersedia="+sedia+" WHERE kode='"+kode+"'";
        PreparedStatement prepare =
konek.prepareStatement(query);
//        prepare.setString(1,kode);
//        prepare.setInt(2, sedia);
        prepare.executeUpdate();
//        JOptionPane.showMessageDialog(null, "Data berhasil
diupdate");
        prepare.close();
    } catch (Exception ex) {
        ex.printStackTrace();
//        JOptionPane.showMessageDialog(null, "Data gagal
diupdate");
    }
}
}
}

```

Lampiran 12

Sourcecode MenuDataset.Java

Nama Activity	Modul
Data set	MenuDataset.Java
<pre> package wana.uke; import java.awt.EventQueue; import javax.swing.JFrame; import javax.swing.JPanel; import javax.swing.border.EmptyBorder; import javax.swing.UIManager; import java.sql.*; import javax.swing.JTextArea; import javax.swing.JScrollPane; @SuppressWarnings("serial") public class MenuDataset extends JFrame { private JPanel contentPane; private JTextArea taMatriks = new JTextArea(); //buat header String hkode[]; int trans=0; int[][] isiDs; int jmlkolom=0; int jmlbaris=0; /** * Launch the application. */ public static void main(String[] args) { EventQueue.invokeLater(new Runnable() { public void run() { try { UIManager.setLookAndFeel("com.jtattoo.plaf.mcwin.McWinLookAndFeel"); MenuDataset frame = new MenuDataset(); frame.setVisible(true); } catch (Exception e) { e.printStackTrace(); } } }); } </pre>	

```

/**
 * Create the frame.
 */
public MenuDataset() {
    setResizable(false);
    setTitle("Data Set Apriori");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 592, 536);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setBounds(10, 11, 556, 476);
    contentPane.add(scrollPane);

    taMatriks.setBounds(10, 11, 566, 476);
    scrollPane.setViewportView(taMatriks);
    tampilMatriks();
} //akhir konstruktur

//tampilkan isi matriks
public void tampilMatriks(){
    getDataTable();
    StringBuilder sb = new StringBuilder();
    StringBuilder sbdata;
    //inisialisasi / cek apakah tabel dataset udah ada?
    inisDataset();

    //ambil data dari matriks dan tampilkan, juga lakukan
    pengisian tabel dataset
    for (int i =0; i <= trans; i++) {
        sbdata = new StringBuilder();
        sb.append("ID"+String.valueOf(i+1)+" = { ");
        for (int j = 0; j < jmlkolom; j++) {
            //untuk tampil
            if(isiDs[i][j]==1){
                sb.append(hkode[j]);
                sb.append(' ');
                //untuk tabel dataset
                sbdata.append(hkode[j]);
            }
        }
        sb.append(" }");
        sb.append('\n');
        tambahDataset(i+1,sbdata.toString());
        //isi tabel dataset
    }
    taMatriks.setText(sb.toString());
}

```

```

private void tambahDataset(int i, String isidata) {
    try {
        //simpan data ke database
        Connection konek = Koneksi.getKoneksi();
        String query = "INSERT INTO dataset VALUES(?,?)";
        PreparedStatement prepare = konek.prepareStatement(query);
        prepare.setInt(1, i);
        prepare.setString(2, isidata);
        prepare.executeUpdate();
        //JOptionPane.showMessageDialog(null, "Data berhasil
ditambahkan ke database");
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}

private void inisDataset() {
    int jumAda=0;
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();

        //ambil jumlah data barang
        String querytmp = "SELECT COUNT(*) AS ada FROM
information_schema.TABLES WHERE table_schema= 'aprioricon' AND
table_name= 'dataset'";
        ResultSet rs = state.executeQuery(querytmp);
        rs.next();
        jumAda = rs.getInt("ada");
        //drop tabel dataset klo ada
        if(jumAda>0){
            String qDrop = "DROP TABLE dataset";
            state.executeUpdate(qDrop);
        }
        //buat tabel dataset
        String qCreate = "CREATE TABLE dataset "+
"(transaksi TINYINT(1) not NULL, "+
" isi VARCHAR("+jmlkolom+"), "+
" PRIMARY KEY ( transaksi ))";
        state.executeUpdate(qCreate);

        rs.close();
        state.close();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
}

//isi tabel

```

```

public void getDataTable()
{
    //ambil jumlah kolom dan isi Afield
    jmlkolom=getKolom();
    jmlbaris=getBaris();
    String kodeHj;
    Date dateHj;
    isiDs = new int[jmlbaris][jmlkolom];
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();
        String query = "SELECT * FROM hisjual ORDER BY
tanggal, kodebrg";
        ResultSet rs = state.executeQuery(query);
        Date datetmp = null;
        int ulang=0;
        while(rs.next())
        {
            dateHj = rs.getDate(1);
            kodeHj = rs.getString(2);

            //cek ketersediaan barang
            if (ceksedia(kodeHj)==1){
                continue;
            }

            //ambil tanggal pertama
            if(ulang==0) {
                datetmp = dateHj;
            }

            //jika tanggal tidak sama
            if(!(datetmp.equals(dateHj))){
                trans++;
                datetmp = dateHj;
            }

            //isi ketersediaan sesuai data barang
            int i=0;
            for(String nkode:hkode){
                if(nkode.equals(kodeHj)){
                    isiDs[trans][i]=1;
                    break;
                }
                i++;
            }
            ulang++;
        }
        rs.close();
        state.close();
    }
    catch(Exception ex)
    {

```

```

        ex.printStackTrace();
    }
}
private int getKolom() {
    int jmlBrs=0;
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();

        //ambil jumlah data barang
        String querytmp = "SELECT COUNT(*) AS baris FROM
barang";

        ResultSet rs = state.executeQuery(querytmp);
        rs.next();
        jmlBrs = rs.getInt("baris");

        //isi matriks hkode dng kode barang
        hkode = new String[jmlBrs];
        String query = "SELECT kode FROM barang";
        rs = state.executeQuery(query);
        int i=0;
        while(rs.next())
        {
            hkode[i] = rs.getString("kode");
            i++;
        }
        rs.close();
        state.close();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
    return jmlBrs;
}

private int getBaris() {
    int jmlBrs=0;
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();

        //ambil jumlah baris tanggal berbeda
        String querytmp = "SELECT COUNT(DISTINCT tanggal) AS
barang FROM hisjual";
        ResultSet rs = state.executeQuery(querytmp);
        rs.next();
        jmlBrs = rs.getInt("baris");
        rs.close();
        state.close();
    }
    catch(Exception ex)

```



```

        {
            ex.printStackTrace();
        }
        return jmlBrs;
    }

    //cek ketersediaan barangsf dan ketidakterediaan barangdistor
    private Byte cekseedia(String kodehis) {
        int jmlsf=0;
        int jmldistor=0;
        byte kembali=0;
        try
        {
            Connection konek = Koneksi.getKoneksi();
            Statement state = konek.createStatement();

            //ambil jumlah ketersediaan barangsf
            String querysf = "SELECT COUNT(*) AS ada FROM
barangsf WHERE kode='"+kodehis+"' AND tersedia=1";
            ResultSet rs = state.executeQuery(querysf);
            rs.next();
            jmlsf = rs.getInt("ada");

            //ambil jumlah ketetersediaan barangsf
            String querydistor = "SELECT COUNT(*) AS tidakada
FROM barangdistor WHERE kode='"+kodehis+"' AND tersedia=0";
            rs = state.executeQuery(querydistor);
            rs.next();
            jmldistor = rs.getInt("tidakada");

            rs.close();
            state.close();
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
        if((jmlsf>=1)|| (jmldistor>=1)){
            kembali = 1;
        }
        return kembali;
    }
}

```

Lampiran 13

Sourcecode MenuProses.Java

Nama Activity	Modul
Proses algoritma apriori dengan <i>constraint</i>	MenuProses.Java
<pre> package wana.uke; import java.awt.EventQueue; import javax.swing.JFrame; import javax.swing.JPanel; import javax.swing.border.EmptyBorder; import javax.swing.JLabel; //import javax.swing.JOptionPane; import javax.swing.JTextField; import javax.swing.JScrollPane; import javax.swing.JButton; import javax.swing.UIManager; import java.awt.event.ActionListener; import java.awt.event.ActionEvent; import java.sql.*; import javax.swing.JTextArea; import javax.swing.SwingConstants; import java.awt.Font; @SuppressWarnings("serial") public class MenuProses extends JFrame { private JPanel contentPane; private JTextField tfMsupp; private JTextField tfConsw1; private JLabel lblSaranStokWeek; private JLabel lblSaranStokWeek_1; private JScrollPane scrollPane_1; private JTextArea taW2; private JTextArea taW1; private JLabel lblSaranStokWeek_2; private JScrollPane scrollPane_2; private JTextArea taW3; String strDataset[]; String strBarang[]; String strc111[]; String strc2item[]; String strAsositem[]; </pre>	

```

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                UIManager.setLookAndFeel("com.jtattoo.plaf.mcwin.McWinLookAndFeel
");
                MenuProses frame = new MenuProses();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */
public MenuProses() {
    setResizable(false);
    setTitle("Proses Algoritma Apriori Constrain Ketat");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 592, 536);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblMinsupp = new JLabel("Minimum Support (%)");
    lblMinsupp.setBounds(10, 11, 123, 14);
    contentPane.add(lblMinsupp);

    tfMsupp = new JTextField();
    tfMsupp.setBounds(158, 8, 40, 20);
    contentPane.add(tfMsupp);
    tfMsupp.setColumns(10);

    JButton btnSimpan = new JButton("Proses");
    btnSimpan.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            getSaranStok();
        }
    });
    btnSimpan.setBounds(277, 11, 255, 41);
    contentPane.add(btnSimpan);

    JLabel lblConstrainWeekI = new JLabel("Constrain PerWeek
(Rp.)");
    lblConstrainWeekI.setBounds(10, 36, 137, 14);

```

```

        contentPane.add(lblConstrainWeekI);

        tfConsw1 = new JTextField();
        tfConsw1.setBounds(157, 33, 86, 20);
        contentPane.add(tfConsw1);
        tfConsw1.setColumns(10);

        lblSaranStokWeek = new JLabel("Kode Barang");
        lblSaranStokWeek.setFont(new Font("Tahoma", Font.PLAIN,
11));

        lblSaranStokWeek.setHorizontalAlignment(SwingConstants.CENTER);
        lblSaranStokWeek.setBounds(45, 119, 97, 14);
        contentPane.add(lblSaranStokWeek);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(60, 138, 70, 344);
        contentPane.add(scrollPane);

        taW1 = new JTextArea();
        scrollPane.setViewportViewView(taW1);

        lblSaranStokWeek_1 = new JLabel("Nama Barang");
        lblSaranStokWeek_1.setFont(new Font("Tahoma", Font.PLAIN,
11));

        lblSaranStokWeek_1.setHorizontalAlignment(SwingConstants.CENTER);
        lblSaranStokWeek_1.setBounds(169, 119, 172, 14);
        contentPane.add(lblSaranStokWeek_1);

        scrollPane_1 = new JScrollPane();
        scrollPane_1.setBounds(144, 137, 250, 345);
        contentPane.add(scrollPane_1);

        taW2 = new JTextArea();
        scrollPane_1.setViewportViewView(taW2);

        lblSaranStokWeek_2 = new JLabel("Harga Barang");
        lblSaranStokWeek_2.setFont(new Font("Tahoma", Font.PLAIN,
11));

        lblSaranStokWeek_2.setHorizontalAlignment(SwingConstants.CENTER);
        lblSaranStokWeek_2.setBounds(407, 119, 106, 14);
        contentPane.add(lblSaranStokWeek_2);

        scrollPane_2 = new JScrollPane();
        scrollPane_2.setBounds(407, 137, 120, 345);
        contentPane.add(scrollPane_2);

        taW3 = new JTextArea();
        scrollPane_2.setViewportViewView(taW3);

        JLabel lblNewLabel = new JLabel("Usulan Barang yang akan
distok oleh Sales Force Tupperware Minggu Ini");

```

```

        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 13));
        lblNewLabel.setBounds(52, 94, 480, 20);
        contentPane.add(lblNewLabel);
    }//akhir konstuktur

    //isi tabel
    public void getSaranStok()
    {
        setArrdataset();
        setC1L1();
        setC2L2();
        setArrasosiasi();
        setSaran();
    }

    private void setArrdataset() {
        try
        {
            Connection konek = Koneksi.getKoneksi();
            Statement state = konek.createStatement();

            //ambil jumlah dataset
            String querytmp = "SELECT COUNT(*) AS baris FROM
dataset";

            ResultSet rs = state.executeQuery(querytmp);
            rs.next();
            int jmlBrs = rs.getInt("baris");

            //isi matriks hkode dng kode barang
            strDataset= new String[jmlBrs];
            String query = "SELECT isi FROM dataset";
            rs = state.executeQuery(query);
            int i=0;
            while(rs.next())
            {
                strDataset[i] = rs.getString("isi");
                i++;
            }
            rs.close();
            state.close();
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
    }

    private void setC1L1() {
        //simpan kode barang ke array
        setArrdatabrg();
        //inisialisasi / cek apakah tabel C1L1 sudah ada
        inisC1L1();
        setC1();
    }

```

```

//menghitung jumlah sesuai minimum support
int persen=getPersensupp();
try
{
    //menghapus itemset yang tidak sesuai dengan minimum
support
    Connection konek = Koneksi.getKoneksi();
    String query = "DELETE FROM c111 WHERE jumlah <
"+persen;
    PreparedStatement prepare =
konek.prepareStatement(query);
    prepare.executeUpdate();
    prepare.close();
}
catch(Exception e)
{
    e.printStackTrace();
}
}

private void setArrdatabrg() {
try
{
    Connection konek = Koneksi.getKoneksi();
    Statement state = konek.createStatement();

//ambil jumlah kodebarang
String querytmp = "SELECT COUNT(*) AS baris FROM
barang";

    ResultSet rs = state.executeQuery(querytmp);
    rs.next();
    int jmlBrs = rs.getInt("baris");

//isi matriks hkode dng kode barang
strBarang= new String[jmlBrs];
String query = "SELECT kode FROM barang";
rs = state.executeQuery(query);
int i=0;
while(rs.next())
{
    strBarang[i] = rs.getString("kode");
    i++;
}
rs.close();
state.close();
}
catch(Exception ex)
{
    ex.printStackTrace();
}
}

```

```

private void inisC111() {
    int jumAda=0;
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();

        //ambil jumlah dataset
        String querytmp = "SELECT COUNT(*) AS ada FROM
information_schema.TABLES WHERE table_schema= 'aprioricon' AND
table_name= 'c111'";
        ResultSet rs = state.executeQuery(querytmp);
        rs.next();
        jumAda = rs.getInt("ada");
        //drop tabel c111 klo ada
        if(jumAda>0){
            String qDrop = "DROP TABLE c111";
            state.executeUpdate(qDrop);
        }
        //buat tabel c111
        String qCreate = "CREATE TABLE c111 "+
        "(itemset VARCHAR(5) not NULL, "+
        " jumlah INTEGER, "+
        " PRIMARY KEY ( itemset ))";
        state.executeUpdate(qCreate);

        rs.close();
        state.close();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
}

private void setC1() {
    int jumBrg=strBarang.length;
    String kodetmp = new String();
    int[] jmltmp = new int[jumBrg];
    int jumKolom=0;
    int i=0;
    int j=0;
    int spasi=0;
    for(String trans:strDataset){
        //ambil jumlah kode setiap baris dataset
        jumKolom=trans.length()/5;
        spasi=0;
        for(i=0;i<jumKolom;i++){
            //ambil setiap kode
            kodetmp = trans.substring(spasi, spasi+5);
            for(j=0;j<jumBrg;j++){
                //hitung jumlah (count) setiap barang
                if(strBarang[j].equals(kodetmp)){

```

```

        jmltmp[j]++;
    }
    }
    spasi+=5;
}
try {
    //simpan C1 ke database
    Connection konek = Koneksi.getKoneksi();
    for(int x=0;x<jumBrg;x++){
        String query = "INSERT INTO c111
VALUES(?,?)";
        PreparedStatement prepare =
konek.prepareStatement(query);
        //mengambil setiap kode pada dataset
        prepare.setString(1, strBarang[x]);
        prepare.setInt(2, jmltmp[x]);
        prepare.executeUpdate();
    }
}
catch (Exception ex) {
    ex.printStackTrace();
}
}

private void setC2L2() {
    //simpan itemset c111 ke array
    setArrc111();
    //inisialisasi / cek apakah tabel C111 sudah ada
    inisC2L2();
    //simpan itemset c212 ke database
    setc212tmp();
    setArritemc2();
    setC2();
    //menghitung jumlah sesuai minimum support
    int persen=getPersensupp();
    try
    {
        //menghapus itemset yang tidak sesuai dengan minimum
support
        Connection konek = Koneksi.getKoneksi();
        String query = "DELETE FROM c212 WHERE jumlah <
"+persen;
        PreparedStatement prepare =
konek.prepareStatement(query);
        prepare.executeUpdate();
        prepare.close();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}

```



```

    }

    private void setArrc111() {
        try
        {
            Connection konek = Koneksi.getKoneksi();
            Statement state = konek.createStatement();

            //ambil jumlah kodebarang
            String querytmp = "SELECT COUNT(*) AS baris FROM
c111";

            ResultSet rs = state.executeQuery(querytmp);
            rs.next();
            int jmlBrs = rs.getInt("baris");

            //isi matriks strc111 dng kode barang
            strc111= new String[jmlBrs];
            String query = "SELECT itemset FROM c111";
            rs = state.executeQuery(query);
            int i=0;
            while(rs.next())
            {
                strc111[i] = rs.getString("itemset");
                i++;
            }
            rs.close();
            state.close();
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
    }

    private void inisC2L2() {
        int jumAda=0;
        try
        {
            Connection konek = Koneksi.getKoneksi();
            Statement state = konek.createStatement();

            //ambil jumlah dataset
            String querytmp = "SELECT COUNT(*) AS ada FROM
information_schema.TABLES WHERE table_schema= 'aprioricon' AND
table_name= 'c2l2'";
            ResultSet rs = state.executeQuery(querytmp);
            rs.next();
            jumAda = rs.getInt("ada");
            //drop tabel c111 klo ada
            if(jumAda>0){
                String qDrop = "DROP TABLE c2l2";
                state.executeUpdate(qDrop);
            }
            //buat tabel c111

```

```

        String qCreate = "CREATE TABLE c212 "+
            "(itemset VARCHAR(10) not NULL, "+
            " jumlah INTEGER, "+
            " PRIMARY KEY ( itemset ))";
        state.executeUpdate(qCreate);

        rs.close();
        state.close();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
}

private void setc212tmp() {
    int j=0;
    int brs=strc111.length;
    String itemtmp= new String();;
    String items = new String();
    for(int i=0;i<brs-1;i++){
        //item1
        itemtmp=strc111[i];
        for(j=i+1;j<brs;j++){
            //gabung item1 dan 2
            items=itemtmp+strc111[j];
            try {
                //simpan item C2 ke database
                Connection konek =
Koneksi.getKoneksi();
                String query = "INSERT INTO c212
VALUES(?,?)";
                PreparedStatement prepare =
konek.prepareStatement(query);
                //simpan setiap itemset ke c212, jumlah
                masih 0
                prepare.setString(1, items);
                prepare.setInt(2,0);
                prepare.executeUpdate();
            }
            catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    }
}

private void setArriitemc2() {
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();
    }
}

```

```

        //ambil jumlah item c2
        String querytmp = "SELECT COUNT(*) AS baris FROM
c212";

        ResultSet rs = state.executeQuery(querytmp);
        rs.next();
        int jmlBrs = rs.getInt("baris");

        //isi matriks stc2item dng itemset c212
        strc2item= new String[jmlBrs];
        String query = "SELECT itemset FROM c212";
        rs = state.executeQuery(query);
        int i=0;
        while(rs.next())
        {
            strc2item[i] = rs.getString("itemset");
            i++;
        }
        rs.close();
        state.close();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
}

private void setC2() {
    int jumis=strc2item.length;
    String kodetmp = new String();
    String kodeis = new String();
    int[] jmltmp = new int[jumis];
    int jumKolom=0;
    int i=0;
    int j=0;
    int ada=0;
    int spasi=0;
    for(i=0;i<jumis;i++){
        //scan kemunculan itemset c2 pada dataset
        for(String trans:strDataset){
            //ambil itemset 1
            kodeis=strc2item[i].substring(0,5);
            jumKolom=trans.length()/5;
            spasi=0;
            ada=0;
            for(j=0;j<jumKolom;j++){
                kodetmp = trans.substring(spasi,
spasi+5);

                if(kodeis.equals(kodetmp)){
                    ada++;
                }
            }
            spasi+=5;
        }
    }
}

```

```

        }
        if(ada==2){
            //hitung jumlah (count) setiap itemset
            jmltmp[i]++;
        }
    }
    try {
        //update data c212 dengan memasukkan jumlah
        Connection konek = Koneksi.getKoneksi();
        String query = "UPDATE c212 SET
jumlah="+jmltmp[i]+" WHERE itemset='"+strc2item[i]+"";
        PreparedStatement prepare =
konek.prepareStatement(query);
        prepare.executeUpdate();
        prepare.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

private int getPersensupp() {
    float ps=0;
    int brs=strDataset.length;
    float inpersen = Float.parseFloat(tfMsupp.getText());
    ps= brs*inpersen/100;
    return (int) ps;
}

private void setArrasosiasi() {
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();

        //ambil jumlah c212
        String querytmp = "SELECT COUNT(*) AS baris FROM
c212";

        ResultSet rs = state.executeQuery(querytmp);
        rs.next();
        int jmlBrs = rs.getInt("baris");

        //isi matriks asosiasi dari
        strAsositem= new String[jmlBrs];
        String query = "SELECT itemset FROM c212 ORDER BY
jumlah DESC";

        rs = state.executeQuery(query);
        int i=0;
        while(rs.next())
        {
            strAsositem[i] = rs.getString("itemset");

```

```

        i++;
    }
    rs.close();
    state.close();
}
catch(Exception ex)
{
    ex.printStackTrace();
}
}

private void setSaran() {
    StringBuilder sbKode = new StringBuilder();
    StringBuilder sbNama = new StringBuilder();
    StringBuilder sbHarga = new StringBuilder();
    //saran per Week
    int i=0;
    //ambil asosiasi item kiri dan kanan
    String kiri = strAsositem[i].substring(0,5);
    String kanan = strAsositem[i].substring(5,10);

    float totalW1 = getHarga(kiri)+getHarga(kanan);
    //isi tampung kode
    String tampung=kiri + kanan;
    float batasW1= Float.parseFloat(tfConsw1.getText());
    int ulang=0;
    int j=0;
    int ada=0;
    int spasi=0;
    String cecktampung;

    //cetak usulan perweek
    sbKode.append(kiri);
    sbNama.append(getNama(kiri));
    sbHarga.append("Rp. "+String.valueOf(getHarga(kiri)));

    sbKode.append('\n');
    sbNama.append('\n');
    sbHarga.append('\n');

    sbKode.append(kanan);
    sbNama.append(getNama(kanan));
    sbHarga.append("Rp. "+String.valueOf(getHarga(kanan)));

    sbKode.append('\n');
    sbNama.append('\n');
    sbHarga.append('\n');

    while(totalW1 < batasW1){
        i++;
        kiri = strAsositem[i].substring(0,5);
        kanan = strAsositem[i].substring(5,10);
        ulang = tampung.length()/5;
        ada=0;
    }
}

```

```

spasi=0;
cektampung="";
for(j=0;j<ulang;j++){
    cektampung=tampung.substring(spasi, spasi+5);
    if(kiri.equals(cektampung)){
        ada=1;
    }
    if(kanan.equals(cektampung)){
        ada=2;
    }
    spasi+=5;
}
if(ada==1){
    tampung+=kanan;
    totalW1+=getHarga(kanan);
    sbKode.append(kanan);
    sbNama.append(getNama(kanan));
    sbHarga.append("Rp.
"+String.valueOf(getHarga(kanan)));

    sbKode.append('\n');
    sbNama.append('\n');
    sbHarga.append('\n');
}
else if(ada==2){
    tampung+=kiri;
    totalW1+=getHarga(kiri);
    sbKode.append(kiri);
    sbNama.append(getNama(kiri));
    sbHarga.append("Rp.
"+String.valueOf(getHarga(kiri)));

    sbKode.append('\n');
    sbNama.append('\n');
    sbHarga.append('\n');
}
else {
    tampung=tampung+kiri+kanan;

    totalW1=totalW1+getHarga(kiri)+getHarga(kanan);

    sbKode.append(kiri);
    sbNama.append(getNama(kiri));
    sbHarga.append("Rp.
"+String.valueOf(getHarga(kiri)));

    sbKode.append('\n');
    sbNama.append('\n');
    sbHarga.append('\n');

    sbKode.append(kanan);
    sbNama.append(getNama(kanan));
    sbHarga.append("Rp.
"+String.valueOf(getHarga(kanan)));

```

```

        sbKode.append('\n');
        sbNama.append('\n');
        sbHarga.append('\n');
    }
}
sbKode.append('\n');
sbNama.append('\n');
sbHarga.append('\n');
sbNama.append("Total Usulan Stok Perweek ");
sbHarga.append("Rp. "+String.valueOf(totalW1));
taW1.setText(sbKode.toString());
taW2.setText(sbNama.toString());
taW3.setText(sbHarga.toString());
}

private float getHarga(String kode) {
    float harga=0;
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();

        //ambil harga sesuai kode
        String query = "SELECT harga FROM barang WHERE kode=
"+kode+"";

        ResultSet rs = state.executeQuery(query);
        rs.next();
        harga = rs.getFloat("harga");
        rs.close();
        state.close();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }

    return harga;
}

private String getNama(String kode) {
    String nama = new String();
    try
    {
        Connection konek = Koneksi.getKoneksi();
        Statement state = konek.createStatement();

        //ambil harga sesuai kode
        String query = "SELECT nama FROM barang WHERE kode=
"+kode+"";

        ResultSet rs = state.executeQuery(query);
        rs.next();
        nama = rs.getString("nama");
    }
}

```

```
        rs.close();
        state.close();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
    return nama;
}
}
```