

RSM Phase II - Feature #622

Delete Tombol Action

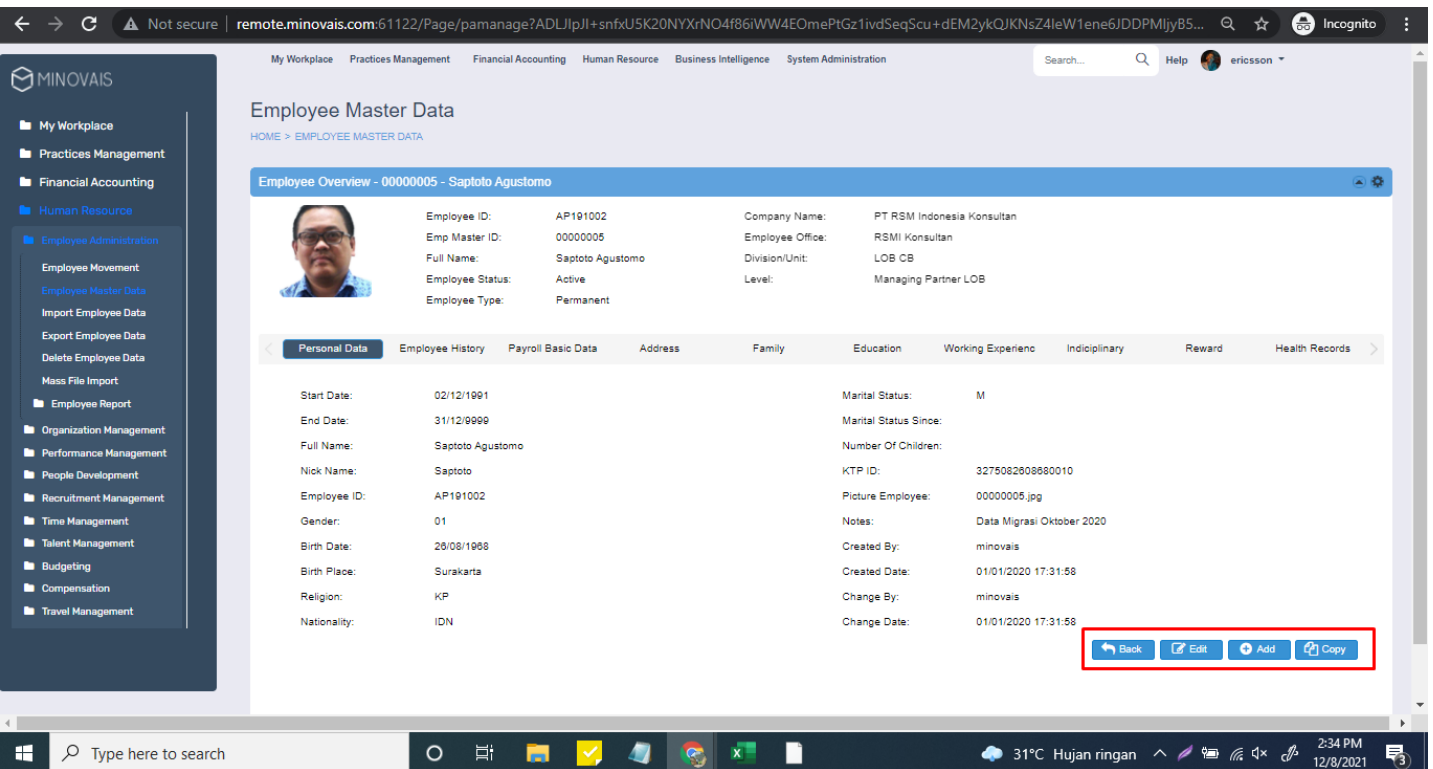
12/08/2021 02:41 PM - Anonymous

Status:	Closed	Start date:	12/08/2021
Priority:	Immediate	Due date:	
Assignee:	ericsson ericsson	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour

Description

Dear, mang wanto.

Mohon bantuannya untuk mendelete tombol action edit, add, copy. jadi hanya ada tombol action back, jika user memiliki roleID dengan menu action 0.



port : 61122
db : rsm prod
server 77 || 2014

History

#1 - 12/08/2021 02:57 PM - Saswanto Tampan

- Status changed from New to QA Test

- Assignee changed from Saswanto Tampan to ericsson ericsson

PADA FILE MD GENERATOR

...\UI\scripts\mdgenerator.js

ada perubahan pada function mdgenerator.createFormCard menjadi seperti di bawah

```
mdgenerator.createFormCard = function (tableName, layout, param_, renderTo, isDisplay, record, act, isfromgrid) {  
  var name = 'MinovaGridFormEdit';  
  var save = true;  
  var edit = false;  
  var delete_ = true;  
  var back = false;  
  var add = false;
```

```

var copy = false;
var cancel = true;
if (isfromgrid == null) isfromgrid = '1';
if (isDisplay) {
    name = 'MinovaGridFormDisplay'
} else {
    save = false;
    edit = true;
    delete_ = false;
    add = true;
    copy = true;
    if (isfromgrid == '1') {
        back = false;
        cancel = true;
    }
    else {
        back = true;
        cancel = false;
    }
}
var userAction = MinovaUtil.GetActionMenu();
if (userAction != '1') {
    Ext.Ajax.request({
        async: false,
        method: 'POST',
        url: '/UserControl/GetStore',
        params: {
            tableName: 'PCUSTROLE',
            param: 'Menu[equal]'+ MinovaUtil.GetMenuID() +',MenuAction[equal]0'
        },
        success: function (response) {
            var results = Ext.decode(response.responseText);
            data_ = results.data;
            if (data_.length > 0) {
                save = true;
                edit = true;
                delete_ = true;
                add = true;
                copy = true;
                cancel = true;
            }else{
                save = true;
                edit = true;
                delete_ = true;
                add = false;
                copy = false;
                cancel = true;
            }
        }
    });
}
if (tableName.toLowerCase().substring(0, 5) == 'phrrc') {
    Ext.Ajax.request({
        async: false,
        method: 'POST',
        url: '/UserControl/GetStore',
        params: {
            tableName: 'phrpa0015',
            param: 'ApplicantID[equal]' + getParam('app')
        },
        success: function (response) {
            var results = Ext.decode(response.responseText);
            data_ = results.data;
            if (data_.length > 0) {
                save = true;
                edit = true;
                delete_ = true;
                add = false;
                copy = false;
                cancel = true;
            }
        }
    });
}
Ext.create('Ext.form.Panel', {

```

```

renderTo: renderTo,
name: name + tableName,
formMD: tableName,
act: null,
items: [{
  xtype: 'minovaForm',
  name: name,
  tableName: tableName,
  margin: '0 0 10 0',
  isDisplay: isDisplay,
  //hidden:true
  buttons: [
    {
      text: 'Save',
      hidden: save,
      act: act,
      //margin: '0 3 15 0', //(top, right, bottom, left).
      name: 'btnSaveMD',
      iconCls: 'fa-save',
      style: 'font-family: FontAwesome',
      handler: function () {
        var p_ = Ext.ComponentQuery.query('[name=Main]')[0];
        //p_.mask('Loading..');
        var frm = Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0];
        var grid_ = Ext.ComponentQuery.query('[gridMD=' + tableName + ']')[0];
        //frm.mask('loading..');
        var panelInt = Ext.ComponentQuery.query('[name=PanelMdDetail]')[0];
        if (panelInt) {
          panelInt.mask('Loading..');
        }

        if (frm.getForm().isValid()) {
          Ext.Ajax.request({
            method: 'POST',
            //async: false,
            url: '/Dev/SaveTable',
            params: {
              tableName: tableName,
              data: Ext.encode(frm.getValues()),
              action: frm.act,
            },
            waitMsg: 'Saving Data...',
            success: function (response) {
              var result = Ext.decode(response.responseText);
              if (result.success) {
                MinovaMessage('Not Null', '000006', '', 'S');
                var menu = Ext.ComponentQuery.query('[name = gridMenuMD]')[0];
                var s = menu.getStore();
                {
                  text: 'Cancel',
                  hidden: cancel,
                  //margin: '0 3 15 0', //(top, right, bottom, left).
                  name: 'btnCancelForm',
                  iconCls: 'fa-undo',
                  style: 'font-family: FontAwesome',
                  viewForm: name + tableName,
                  handler: function () {
                    MinovaMessage('Message', '000011', '', 'C', function (response) {
                      var layout = null;
                      var pnl = Ext.ComponentQuery.query('[name=panelTab' + tableName + ']')[0];
                      if (pnl) {
                        layout = pnl.layoutType;
                      }
                      if (response == "yes") {
                        //if (layout == "L004") {
                          Ext.ComponentQuery.query('[name=' + name + tableName + ']')[0].setHidden(true)
                        }
                      }
                      var FormEdit = Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName
+ ']')[0];
                      if (FormEdit) {
                        FormEdit.destroy();
                      }
                      var FormDis = Ext.ComponentQuery.query('[name=MinovaGridFormDisplay' + tableNa
me + ']')[0];
                      if (FormDis) {

```

```

        FormDis.setHidden(false);
    }
    //}
}
});
{
    text: ' Delete ',
    hidden: delete_,
    //margin: ' 0 3 15 0 ', //(top, right, bottom, left).
    iconCls: ' fa-trash-o ',
    name: ' btnDeleteMD ',
    style: ' font - family : FontAwesome ',
    action: ' CRUDdelete ',
    handler: function () {
{
    text: 'Back',
    hidden: back,
    //margin: '0 3 15 0', //(top, right, bottom, left).
    name: 'btnCancelMD',
    iconCls: 'fa-reply',
    style: 'font-family: FontAwesome',
    viewForm: name + tableName,
    handler: function () {
        MinovaMessage('Message', '000011', '', 'C', function (response) {
            if (response == "yes") {
                Ext.ComponentQuery.query('[name=' + name + tableName + ''])[0].setHidden(true)
;
                Ext.ComponentQuery.query('[gridMD=' + tableName + ''])[0].setHidden(false);
                var p = Ext.ComponentQuery.query('[name=PanelMdDetail]')[0];
                if (Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ''])[0])
                {
                    Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ''])[0].
act = null;
                }
                var grid_ = Ext.ComponentQuery.query('minovagridmd[tableName=' + tableName + '
]')[0];
                Ext.ComponentQuery.query('[name=DetailMD]')[0].setTitle(grid_.tableDesc + ' -
' + getParam("EmployeeID") + ' - ' + getParam("FullName"));
                p.setHeight(460);
                var FormEdit = Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName
+ '']')[0];
                if (FormEdit) {
                    FormEdit.destroy();
                }
                var FormDis = Ext.ComponentQuery.query('[name=MinovaGridFormDisplay' + tableNa
me + '']')[0];
                if (FormDis) {
                    FormDis.destroy();
                }
            }
        });
    }
},
{
    text: 'Edit',
    hidden: edit,
    record: record,
    //margin: '0 3 15 0', //(top, right, bottom, left).
    //name: 'btnEditMD',
    iconCls: 'fa-edit',
    style: 'font-family: FontAwesome',
    handler: function () {
        var p_ = Ext.ComponentQuery.query('[name=Main]')[0];
        p_.mask('Loading...');
        var tbl = tableName;
        var grid_ = Ext.ComponentQuery.query('minovagridmd[tableName=' + tableName + '']')[0];
        var selecton_ = grid_.getView().getSelectionModel().getSelection()[0];
        renderToForm = Ext.ComponentQuery.query('[gridMD=' + tableName + ''])[0].renderToForm;
        if (Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + '']')[0]) {
            Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + '']')[0].setHidde
n(false)
        } else {
            mdgenerator.createFormCard(tableName, "", "", renderToForm, false, selecton_, '1',
'0');
        }
    }
}
}

```

```

{
    text: 'Add',
    hidden: add,
    record: record,
    //margin: '0 3 15 0', //(top, right, bottom, left).
    //name: 'btnAddMD',
    iconCls: 'fa-plus-circle',
    style: 'font-family: FontAwesome',
    handler: function () {
        var p_ = Ext.ComponentQuery.query('[name=Main]')[0];
        p_.mask('Loading...');
        var grid_ = Ext.ComponentQuery.query('minovagridmd[tableName=' + tableName + ']')[0];
        renderToForm = Ext.ComponentQuery.query('[gridMD=' + tableName + ']')[0].renderToForm;
        if (Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0]) {
            Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0].setHidden(false)
        } else {
            mdgenerator.createFormCard(tableName, "", "", renderToForm, false, '', '0', '0');
        }
        var form_ = Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0].
getForm();
        form_.reset();
        form_.findField('EmployeeID').setValue(getParam('EmployeeID'));
        var p = Ext.ComponentQuery.query('[name=PanelMdDetail]')[0];
        var tinggi = Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0].
.getHeight();
        if (tinggi < 330) {
            tinggi = 330;
        }
        p.setHeight(tinggi + 100);
        Ext.ComponentQuery.query('[gridMD=' + tableName + ']')[0].setHidden(true);
        Ext.ComponentQuery.query('[name=MinovaGridFormDisplay' + tableName + ']')[0].setHidden(true); //Bedanya disini
        Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0].act = '0';
        Ext.ComponentQuery.query('[name=DetailMD]')[0].setTitle('Add' + ' - ' + grid_.tableDesc + ' - ' + getParam("EmployeeID") + ' - ' + getParam("FullName"));
        p_.unmask();
        setKeyDisabled(false);
    }
},
{
    text: 'Copy',
    hidden: copy,
    record: record,
    //margin: '0 3 15 0', //(top, right, bottom, left).
    //name: 'btnCopyMD',
    iconCls: 'fa-copy',
    style: 'font-family: FontAwesome',
    handler: function () {
        var grid_ = Ext.ComponentQuery.query('[gridMD=' + tableName + ']')[0];
        var selecton_ = grid_.getView().getSelectionModel().getSelection()[0];
        if (selecton_) {
            var p_ = Ext.ComponentQuery.query('[name=Main]')[0];
            p_.mask('Loading...');
            renderToForm = Ext.ComponentQuery.query('[gridMD=' + tableName + ']')[0].renderToForm;
orm;
            if (Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0]) {
                Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0].setHidden(false)
            } else {
                mdgenerator.createFormCard(tableName, "", "", renderToForm, false, selecton_, '0', '0');
            }
            Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0].getForm().reset();
            Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0].getForm().setValues(selecton_.data);
            var p = Ext.ComponentQuery.query('[name=PanelMdDetail]')[0];
            var tinggi = Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ']')[0].getHeight();
            if (tinggi < 330) {
                tinggi = 330;
            }
            p.setHeight(tinggi + 100);
            grid_.setHidden(true);

```

```

        Ext.ComponentQuery.query('[name=MinovaGridFormDisplay' + tableName + ''])[0].setHidden(true); //Bedanya disini
        Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ''])[0].act = '0';
        Ext.ComponentQuery.query('[name=DetailMD]')[0].setTitle('Copy' + ' - ' + grid_.tableDesc + ' - ' + getParam("EmployeeID") + ' - ' + getParam("FullName"));
        var tbl = tableName;
        setValueAmountEncrypt(tbl);
        p_.unmask();
        setKeyDisabled(false);
        setFileID(0);
    }
},
],
},
]
});

Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ''])[0].getForm().setValues(selecton_.data);
var p = Ext.ComponentQuery.query('[name=PanelMdDetail]')[0];
var tinggi = Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ''])[0].getHeight();

if (tinggi < 330) {
    tinggi = 330;
}
p.setHeight(tinggi + 100);
Ext.ComponentQuery.query('[name=MinovaGridFormDisplay' + tableName + ''])[0].setHidden(true);

Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ''])[0].act = '1';
//Ext.ComponentQuery.query('[name=MinovaGridFormEdit' + tableName + ''])[0].setTitle('Edit - ' + grid_.tableDesc + ' - ' + getParam("EmployeeID") + ' - ' + getParam("FullName"));
Ext.ComponentQuery.query('[name=DetailMD]')[0].setTitle('Edit' + ' - ' + grid_.tableDesc + ' - ' + getParam("EmployeeID") + ' - ' + getParam("FullName"));

setValueAmountEncrypt(tbl);
p_.unmask();
setKeyDisabled(true);
},
},

Ext.ComponentQuery.query('[name=Main]')[0].lastView = Ext.ComponentQuery.query('[name=' + name + tableName + '])[0];
}

```

=====

perubahan text pada tombol Back to Grid
pada page

...\app\view\masterdata\pa\main.js

ada perubahan pada komponen buttons menjadi seperti di bawah

```

text: 'Back',
hidden: true,
margin: '0 3 15 0', //(top, right, bottom, left).
name: 'btnCancelMD',
iconCls: 'fa-reply',
style: 'font-family: FontAwesome',

```

#2 - 12/09/2021 10:11 AM - ericsson ericsson

- Status changed from QA Test to Closed

Files

File Name	Size	Date	Author
clipboard-202112081439-ku2y5.png	153 KB	12/08/2021	Anonymous