## PROGRAM DEFINITION

|  |
| --- |
|  |
| * A program to generate a report for a list of overtime charges that already or will be processed into an invoice |

## RELATED TABLES

|  |  |
| --- | --- |
|  | **Access** |
| * - | * - |

## ROLES

|  |  |
| --- | --- |
|  | **Test** |
| * User that has access to program |  |

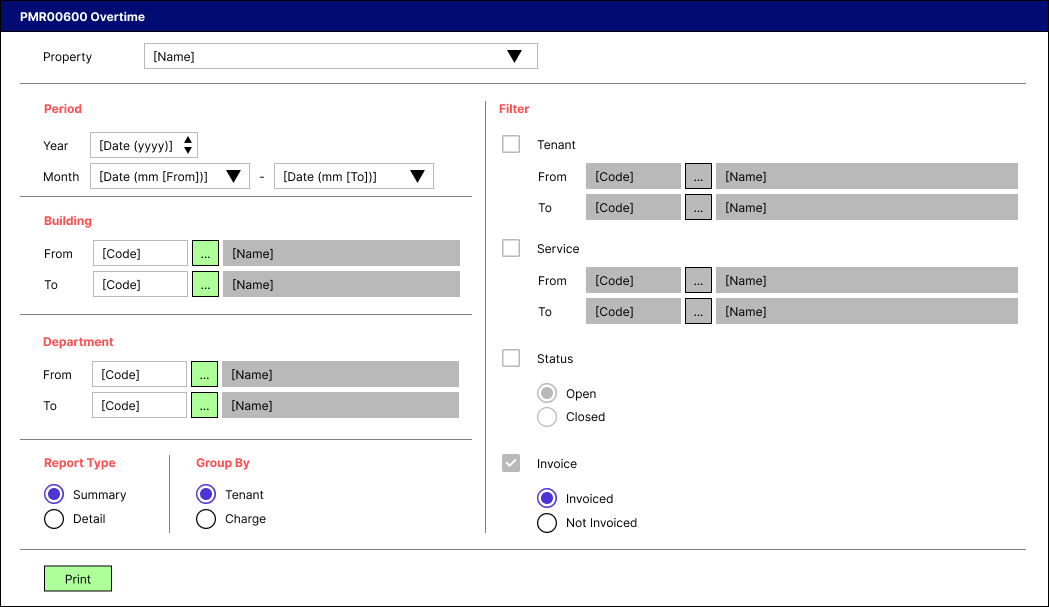
## INITIAL PROCESS

|  |  |
| --- | --- |
|  | **Test** |
| * Set VAR\_COMPANY\_ID = Login User’s Company ID * Set VAR\_USER\_ID = Login User’s ID * Set VAR\_LANGUAGE\_ID = Login User’s Language ID * Execute this stored procedure, save the results to VAR\_YEAR\_RANGE   EXEC RSP\_GS\_GET\_PERIOD\_YEAR\_RANGE  VAR\_COMPANY\_ID  , ''  , ''   * Set VAR\_LOOKUP\_BUILDING\_LIST\_GET\_RECORD = Value from lookup GSL02200 * Set VAR\_LOOKUP\_DEPARTMENT\_LIST\_GET\_RECORD = Value from lookup GSL00700 * Set VAR\_LOOKUP\_TENANT\_LIST\_GET\_RECORD = Value from lookup PML00600 * Set VAR\_LOOKUP\_SERVICE\_LIST\_GET\_RECORD = Value from lookup PML00400 |  |

## MAIN FORM

### USER INTERFACE

#### INITIAL VERSION



### BUTTON DEFINITION

|  |  |
| --- | --- |
| **Button Name** | **Action** |
| Print | Run PRINT REPORT PROCESS |

### DISPLAY PROCESS

#### FIELD DEFINITION

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Field** | **Type** | **Validation** |
| Property |  | Combo box  Set VAR\_PROPERTY\_LIST with  EXEC RSP\_GS\_GET\_PROPERTY\_LIST  VAR\_COMPANY\_ID  , VAR\_USER\_ID  Set property id = CPROPERTY\_ID  Set property name = CPROPERTY\_NAME  Set default selected property id = top 1 property id from VAR\_PROPERTY\_LIST.  Show name from the specific property id | * Mandatory * - |
| Period [Year] |  | Spinner  Spinner value:   * Minimum = VAR\_YEAR\_RANGE.IMIN\_YEAR * Maximum = VAR\_YEAR\_RANGE.IMAX\_YEAR   Set default spinner value:  SELECT YEAR(dbo.RFN\_GET\_DB\_TODAY (VAR\_COMPANY\_ID)) | * Mandatory * - |
| Period [Month [From]] |  | Combo box  Set VAR\_PERIOD\_MONTH\_LIST (as array of object) with VAR\_MONTH\_NUMBER and VAR\_MONTH\_NAME.  Set VAR\_MONTH\_NUMBER = 01, 02, … , 12  Set VAR\_MONTH\_NAME = January, February, … , December  Set default combo box value month name based on  SELECT MONTH(dbo.RFN\_GET\_DB\_TODAY (VAR\_COMPANY\_ID)) | * Mandatory * - |
| Period [Month [To]] |  | Combo box  Set VAR\_PERIOD\_MONTH\_LIST (as array of object) with VAR\_MONTH\_NUMBER and VAR\_MONTH\_NAME.  Set VAR\_MONTH\_NUMBER = 01, 02, … , 12  Set VAR\_MONTH\_NAME = January, February, … , December  Set default combo box value month name based on  SELECT MONTH(dbo.RFN\_GET\_DB\_TODAY (VAR\_COMPANY\_ID)) | * Mandatory * - |
| Building [From [Code]] |  | Text box | * Mandatory * If user inputs manually, then check its name based on VAR\_LOOKUP\_BUILDING\_LIST\_GET\_RECORD and save it to VAR\_BUILDING\_NAME. Finally: * If VAR\_BUILDING\_NAME is empty, then: * Display message “Building [INPUT Building Code] not found!”, [OK] * Set focus to Building [From [Code]] field * Else, set Building [From [Name]]’s value = VAR\_BUILDING\_NAME * - |
| Building [From [Lookup]] |  | Lookup | * Enter lookup program GSL02200 with parameters: VAR\_COMPANY\_ID, selected Property ID, VAR\_USER\_ID, 0. Then save the result to VAR\_BUILDING\_RESULT * If VAR\_BUILDING\_RESULT is empty, then: * Display message “Building [INPUT Building Code] not found!”, [OK] * Set focus to Building [From [Code]] field * Else if VAR\_BUILDING\_RESULT exists, then: * Set Building [From [Code]] = CBUILDING\_ID * Set Building [From [Name]] = CBUILDING\_NAME * - |
| Building [From [Name]] |  | Text box | * Disabled * - |
| Building [To [Code]] |  | Text box | * Mandatory * If user inputs manually, then check its name based on VAR\_LOOKUP\_BUILDING\_LIST\_GET\_RECORD and save it to VAR\_BUILDING\_NAME. Finally: * If VAR\_BUILDING\_NAME is empty, then: * Display message “Building [INPUT Building Code] not found!”, [OK] * Set focus to Building [To [Code]] field * Else, set Building [To [Name]]’s value = VAR\_BUILDING\_NAME * - |
| Building [To [Lookup]] |  | Lookup | * Enter lookup program GSL02200 with parameters: VAR\_COMPANY\_ID, selected Property ID, VAR\_USER\_ID, 0. Then save the result to VAR\_BUILDING\_RESULT * If VAR\_BUILDING\_RESULT is empty, then: * Display message “Building [INPUT Building Code] not found!”, [OK] * Set focus to Building [To [Code]] field * Else if VAR\_BUILDING\_RESULT exists, then: * Set Building [To [Code]] = CBUILDING\_ID * Set Building [To [Name]] = CBUILDING\_NAME * - |
| Building [To [Name]] |  | Text box | * Disabled * - |
| Department [From [Code]] |  | Text box | * Mandatory * If user inputs manually, then check its name based on VAR\_LOOKUP\_DEPARTMENT\_LIST\_GET\_RECORD and save it to VAR\_DEPARTMENT\_NAME. Finally: * If VAR\_DEPARTMENT\_NAME is empty, then: * Display message “Department [INPUT Department Code] not found!”, [OK] * Set focus to Department [From [Code]] field * Else, set Department [From [Name]]’s value = VAR\_DEPARTMENT\_NAME * - |
| Department [From [Lookup]] |  | Lookup | * Enter lookup program GSL00700 with parameters: VAR\_COMPANY\_ID, VAR\_USER\_ID. Then save the result to VAR\_DEPARTMENT\_RESULT * If VAR\_DEPARTMENT\_RESULT is empty, then: * Display message “Department [INPUT Department Code] not found!”, [OK] * Set focus to Department [From [Code]] field * Else if VAR\_DEPARTMENT\_RESULT exists, then: * Set Department [From [Code]] = CDEPT\_CODE * Set Department [From [Name]] = CDEPT\_NAME * - |
| Department [From [Name]] |  | Text box | * Disabled * - |
| Department [To [Code]] |  | Text box | * Mandatory * If user inputs manually, then check its name based on VAR\_LOOKUP\_DEPARTMENT\_LIST\_GET\_RECORD and save it to VAR\_DEPARTMENT\_NAME. Finally: * If VAR\_DEPARTMENT\_NAME is empty, then: * Display message “Department [INPUT Department Code] not found!”, [OK] * Set focus to Department [To [Code]] field * Else, set Department [To [Name]]’s value = VAR\_DEPARTMENT\_NAME * - |
| Department [To [Lookup]] |  | Lookup | * Enter lookup program GSL00700 with parameters: VAR\_COMPANY\_ID, VAR\_USER\_ID. Then save the result to VAR\_DEPARTMENT\_RESULT * If VAR\_DEPARTMENT\_RESULT is empty, then: * Display message “Department [INPUT Department Code] not found!”, [OK] * Set focus to Department [To [Code]] field * Else if VAR\_DEPARTMENT\_RESULT exists, then: * Set Department [To [Code]] = CDEPT\_CODE * Set Department [To [Name]] = CDEPT\_NAME * - |
| Department [To [Name]] |  | Text box | * Disabled * - |
| Report Type |  | Radio button  Radio Value   * ‘1’ = Summary * ‘2’ = Detail   Set default value to Summary | * Mandatory * If selected value = Summary, then: * Set Group By’s value = Tenant * Disable Filter [Tenant] * Set Filter [Tenant]’s value = False * Disable Filter [Service] * Set Filter [Service]’s value = False * Disable Filter [Status] * Set Filter [Status]’s value = False * Disable Filter [Invoice] * Set Filter [Invoice]’s value = True * Else: * Enable Filter [Tenant] * Enable Filter [Service] * Enable Filter [Status] * Enable Filter [Invoice] * Set Filter [Invoice]’s value = False * - |
| Group By |  | Radio button  Radio Value   * ‘1’ = Tenant * ‘2’ = Charge   Set default value to Tenant | * Mandatory * - |
| Filter [Tenant] |  | Check box  Set:   * When True, then 1 * When False, then 0   Set default value to False | * Optional * If selected value = True, then enable: * Enable Filter [Tenant [From [Code]]] * Enable Filter [Tenant [From [Lookup]]] * Enable Filter [Tenant [To [Code]]] * Enable Filter [Tenant [To [Lookup]]] * Else if selected value = False, then disable: * Disable Filter [Tenant [From [Code]]] * Set Filter [Tenant [From [Code]]]’s value to empty * Disable Filter [Tenant [From [Lookup]]] * Disable Filter [Tenant [To [Code]]] * Set Filter [Tenant [To [Code]]]’s value to empty * Disable Filter [Tenant [To [Lookup]]] * - |
| Filter [Tenant [From [Code]]] |  | Text box | * Mandatory, if Filter [Tenant]’s value = True * If user inputs manually, then check its name based on VAR\_LOOKUP\_TENANT\_LIST\_GET\_RECORD and save it to VAR\_TENANT\_NAME. Finally: * If VAR\_TENANT\_NAME is empty, then: * Display message “Tenant [INPUT Tenant Code] not found!”, [OK] * Set focus to Filter [Tenant [From [Code]]] field * Else, set Filter [Tenant [From [Name]]]’s value = VAR\_TENANT\_NAME * - |
| Filter [Tenant [From [Lookup]]] |  | Lookup | * Enter lookup program PML00600 with parameters: VAR\_COMPANY\_ID, selected Property ID, “01”, VAR\_USER\_ID. Then save the result to VAR\_TENANT\_RESULT * If VAR\_TENANT\_RESULT is empty, then: * Display message “Tenant [INPUT Tenant Code] not found!”, [OK] * Set focus to Filter [Tenant [From [Code]]] field * Else if VAR\_TENANT\_RESULT exists, then: * Set Filter [Tenant [From [Code]]] = CTENANT\_ID * Set Filter [Tenant [From [Name]]] = CTENANT\_NAME * - |
| Filter [Tenant [From [Name]]] |  | Text box | * Disabled * - |
| Filter [Tenant [To [Code]]] |  | Text box | * Mandatory, if Filter [Tenant]’s value = True * If user inputs manually, then check its name based on VAR\_LOOKUP\_TENANT\_LIST\_GET\_RECORD and save it to VAR\_TENANT\_NAME. Finally: * If VAR\_TENANT\_NAME is empty, then: * Display message “Tenant [INPUT Tenant Code] not found!”, [OK] * Set focus to Filter [Tenant [To [Code]]] field * Else, set Filter [Tenant [To [Name]]]’s value = VAR\_TENANT\_NAME * - |
| Filter [Tenant [To [Lookup]]] |  | Lookup | * Enter lookup program PML00600 with parameters: VAR\_COMPANY\_ID, selected Property ID, “01”, VAR\_USER\_ID. Then save the result to VAR\_TENANT\_RESULT * If VAR\_TENANT\_RESULT is empty, then: * Display message “Tenant [INPUT Tenant Code] not found!”, [OK] * Set focus to Filter [Tenant [To [Code]]] field * Else if VAR\_TENANT\_RESULT exists, then: * Set Filter [Tenant [To [Code]]] = CTENANT\_ID * Set Filter [Tenant [To [Name]]] = CTENANT\_NAME * - |
| Filter [Tenant [To [Name]]] |  | Text box | * Disabled * - |
| Filter [Service] |  | Check box  Set:   * When True, then 1 * When False, then 0   Set default value to False | * Optional * If selected value = True, then enable: * Enable Filter [Service [From [Code]]] * Enable Filter [Service [From [Lookup]]] * Enable Filter [Service [To [Code]]] * Enable Filter [Service [To [Lookup]]] * Else if selected value = False, then disable: * Disable Filter [Service [From [Code]]] * Set Filter [Service [From [Code]]]’s value to empty * Disable Filter [Service [From [Lookup]]] * Disable Filter [Service [To [Code]]] * Set Filter [Service [To [Code]]]’s value to empty * Disable Filter [Service [To [Lookup]]] * - |
| Filter [Service [From [Code]]] |  | Text box | * Mandatory, if Filter [Service]’s value = True * If user inputs manually, then check its name based on VAR\_LOOKUP\_SERVICE\_LIST\_GET\_RECORD and save it to VAR\_SERVICE\_NAME. Finally: * If VAR\_SERVICE\_NAME is empty, then: * Display message “Service [INPUT Service Code] not found!”, [OK] * Set focus to Filter [Service [From [Code]]] field * Else, set Filter [Service [From [Name]]]’s value = VAR\_SERVICE\_NAME * - |
| Filter [Service [From [Lookup]]] |  | Lookup | * Enter lookup program PML00400 with parameters: VAR\_COMPANY\_ID, selected Property ID, “08”, VAR\_USER\_ID. Then save the result to VAR\_SERVICE\_RESULT * If VAR\_SERVICE\_RESULT is empty, then: * Display message “Service [INPUT Service Code] not found!”, [OK] * Set focus to Filter [Service [From [Code]]] field * Else if VAR\_SERVICE\_RESULT exists, then: * Set Filter [Service [From [Code]]] = CCHARGES\_ID * Set Filter [Service [From [Name]]] = CCHARGES\_NAME * - |
| Filter [Service [From [Name]]] |  | Text box | * Disabled * - |
| Filter [Service [To [Code]]] |  | Text box | * Mandatory, if Filter [Service]’s value = True * If user inputs manually, then check its name based on VAR\_LOOKUP\_SERVICE\_LIST\_GET\_RECORD and save it to VAR\_SERVICE\_NAME. Finally: * If VAR\_SERVICE\_NAME is empty, then: * Display message “Service [INPUT Service Code] not found!”, [OK] * Set focus to Filter [Service [To [Code]]] field * Else, set Filter [Service [To [Name]]]’s value = VAR\_SERVICE\_NAME * - |
| Filter [Service [To [Lookup]]] |  | Lookup | * Enter lookup program PML00400 with parameters: VAR\_COMPANY\_ID, selected Property ID, “08”, VAR\_USER\_ID. Then save the result to VAR\_SERVICE\_RESULT * If VAR\_SERVICE\_RESULT is empty, then: * Display message “Service [INPUT Service Code] not found!”, [OK] * Set focus to Filter [Service [To [Code]]] field * Else if VAR\_SERVICE\_RESULT exists, then: * Set Filter [Service [To [Code]]] = CCHARGES\_ID * Set Filter [Service [To [Name]]] = CCHARGES\_NAME * - |
| Filter [Service [To [Name]]] |  | Text box | * Disabled * - |
| Filter [Status] |  | Check box  Set:   * When True, then 1 * When False, then 0   Set default value to False | * Optional * If selected value = True, then enable: * Enable Filter [Status [Sub]] * Else if selected value = False, then disable: * Disable Filter [Status [Sub]] * - |
| Filter [Status [Sub]] |  | Radio button  Radio Value   * ‘1’ = Open * ‘2’ = Closed   Set default value to Open | * Mandatory, if Filter [Status]’s value = True * - |
| Filter [Invoice] |  | Check box  Set:   * When True, then 1 * When False, then 0   Set default value to False | * Optional * If selected value = True, then enable: * Enable Filter [Invoice [Sub]] * Else if selected value = False, then disable: * Disable Filter [Invoice [Sub]] * - |
| Filter [Invoice [Sub]] |  | Radio button  Radio Value   * ‘1’ = Open * ‘2’ = Closed   Set default value to Open | * Mandatory, if Filter [Invoice]’s value = True * - |

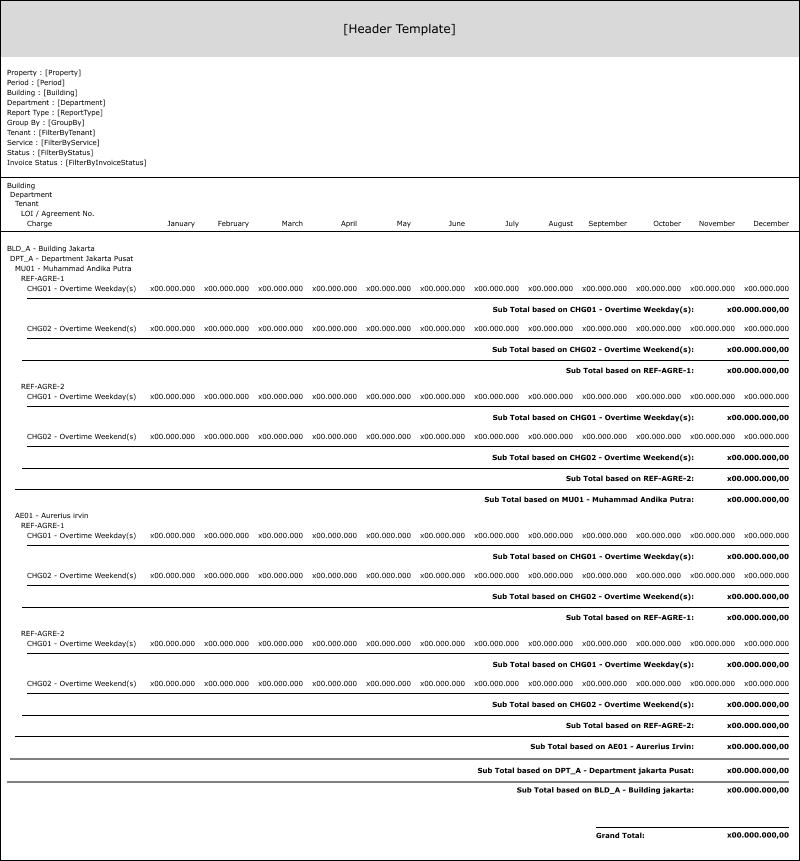
### PRINT REPORT PROCESS

#### REPORT LAYOUT

REPORT TYPE: SUMMARY && GROUP BY: TENANT

PAPER SIZE: A4 PORTRAIT

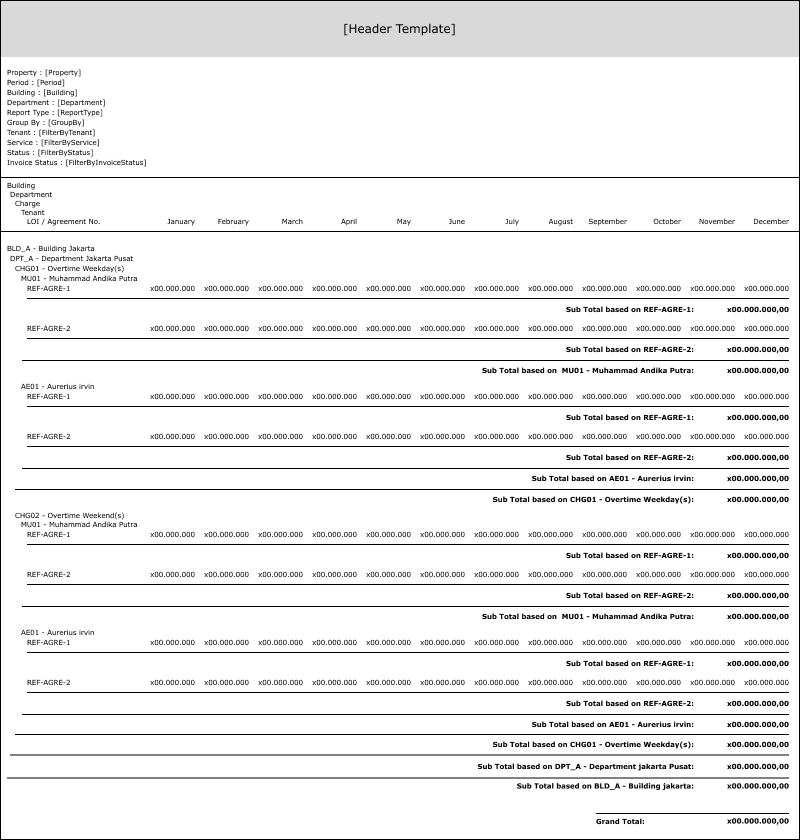
FONT SIZE: 7



REPORT TYPE: SUMMARY && GROUP BY: CHARGE

PAPER SIZE: A4 PORTRAIT

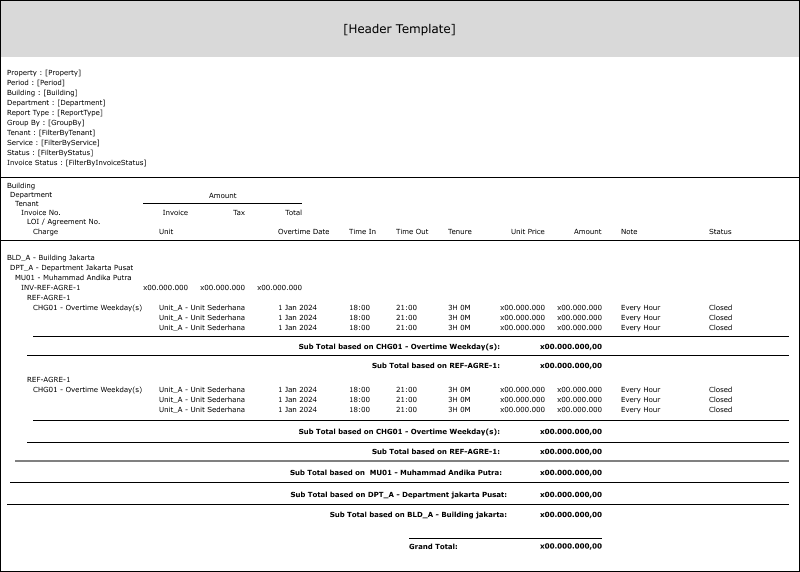
FONT SIZE: 7



REPORT TYPE: DETAIL && GROUP BY: TENANT

PAPER SIZE: A4 PORTRAIT

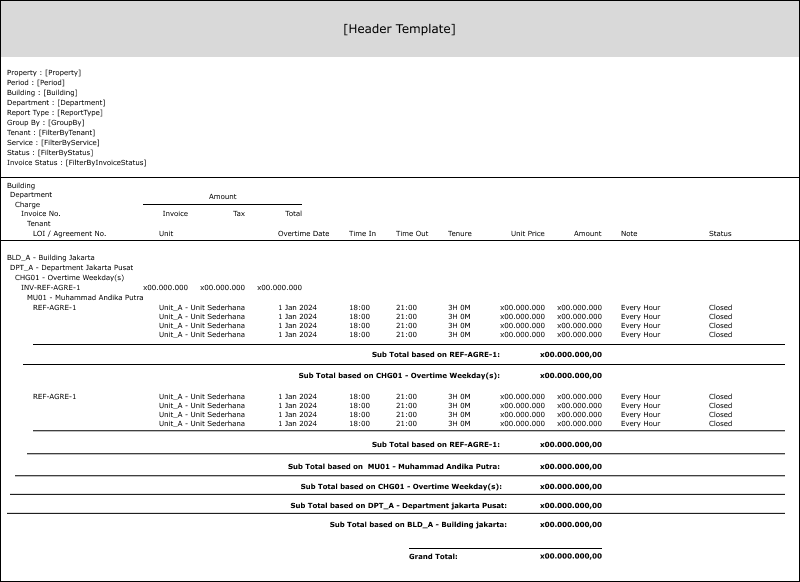
FONT SIZE: 7



REPORT TYPE: DETAIL && GROUP BY: CHARGE

PAPER SIZE: A4 PORTRAIT

FONT SIZE: 7



#### REPORT DEFINITION

GROUP BY TENANT

|  |  |  |  |
| --- | --- | --- | --- |
| **Report** | **Name** | **Field** | **Validation** |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |

GROUP BY CHARGE

|  |  |  |  |
| --- | --- | --- | --- |
| **Report** | **Name** | **Field** | **Validation** |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |

#### VALIDATION

|  |  |  |
| --- | --- | --- |
| **Field** | **Condition** | **Validation Message** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

#### ACTION

|  |
| --- |
|  |
| * Execute this stored procedure: * - |

# CR LIST

|  |  |  |
| --- | --- | --- |
| **CR** | **Description** | **Notes** |
|  |  |  |

#### TEMPLATE

#### VALIDATION

|  |  |  |
| --- | --- | --- |
| **Field** | **Condition** | **Validation Message** |
|  |  |  |
|  |  |  |

#### ACTION

|  |
| --- |
|  |
|  |