## PROGRAM DEFINITION

|  |
| --- |
|  |
| * A program to print a report for a list of LOCs that need to be followed up on or have succeeded in becoming Letter of Intent (LOI) |

## RELATED TABLES

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

## ROLES

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

## INITIAL PROCESS

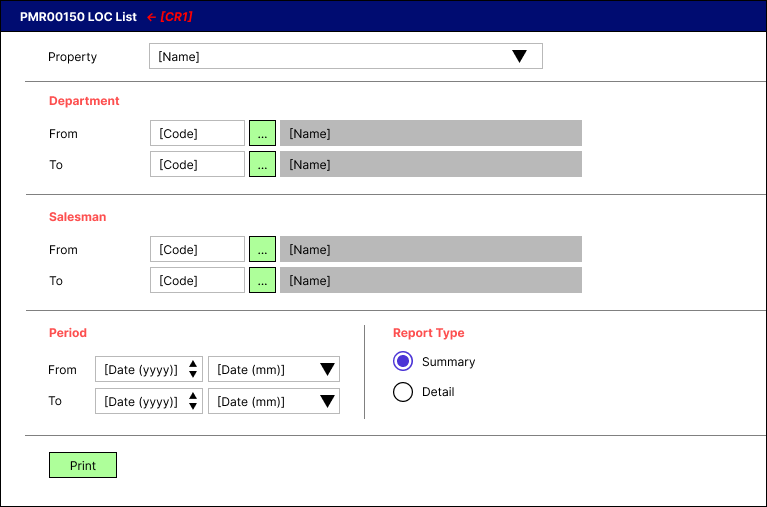
|  |  |
| --- | --- |
|  | **Test** |
| * Set VAR\_COMPANY\_ID = Login User’s Company ID * Set VAR\_USER\_ID = Login User ID * Set VAR\_LANGUAGE\_ID = Login User’s Language ID * Execute this stored procedure, save the results to VAR\_YEAR\_RANGE   EXEC RSP\_GS\_GET\_PERIOD\_YEAR\_RANGE  VAR\_COMPANY\_ID  , ''  , ''   * Set VAR\_LOOKUP\_DEPARTMENT\_LIST\_GET\_RECORD = Value from lookup GSL00700 with the following parameters: * VAR\_COMPANY\_ID * VAR\_USER\_ID * ‘’ * Set VAR\_LOOKUP\_SALESMAN\_LIST\_GET\_RECORD = Value from lookup PML00500 with the following parameters: * VAR\_COMPANY\_ID * selected Property ID * VAR\_USER\_ID * - |  |

## MAIN FORM

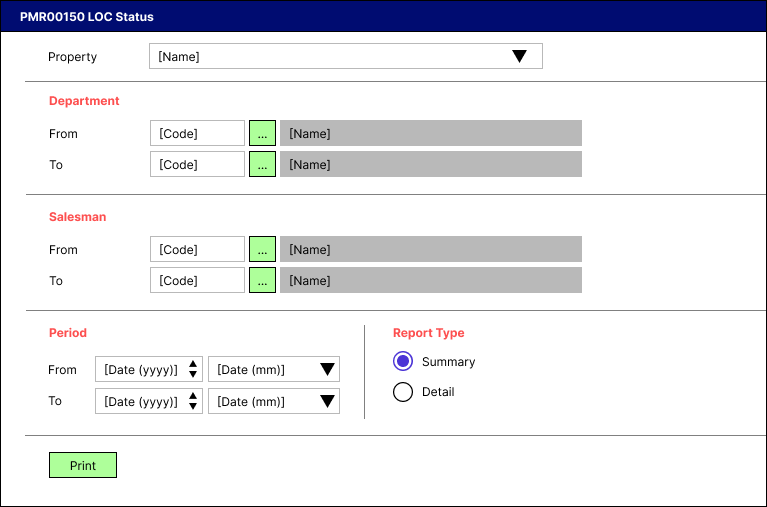
### USER INTERFACE

#### INITIAL VERSION

**[--CR1]**



**[--/CR1]**



### 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 * - |
| 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 * - |
| Salesman [From [Code]] |  | Text box | * Mandatory * If user inputs manually, then check its name based on VAR\_LOOKUP\_SALESMAN\_LIST\_GET\_RECORD and save it to VAR\_SALESMAN\_NAME. Finally: * If VAR\_SALESMAN\_NAME is empty, then: * Display message “Salesman [INPUT Salesman Code] not found!”, [OK] * Set focus to Salesman [From [Code]] field * Else, set Salesman [From [Name]]’s value = VAR\_SALESMAN\_NAME * - |
| Salesman [From [Lookup]] |  | Lookup | * Enter lookup program PML00500 with parameters: * VAR\_COMPANY\_ID * Selected Property Id * VAR\_USER\_ID   Then save the result to VAR\_SALESMAN\_RESULT   * If VAR\_SALESMAN\_RESULT is empty, then: * Display message “Salesman [INPUT Salesman Code] not found!”, [OK] * Set focus to Salesman [From [Code]] field * Else if VAR\_SALESMAN\_RESULT exists, then: * Set Salesman [From [Code]] = CSALESMAN\_ID * Set Salesman [From [Name]] = CSALESMAN\_NAME * - |
| Salesman [From [Name]] |  | Text box | * Disabled * - |
| Salesman [To [Code]] |  | Text box | * Mandatory * If user inputs manually, then check its name based on VAR\_LOOKUP\_SALESMAN\_LIST\_GET\_RECORD and save it to VAR\_SALESMAN\_NAME. Finally: * If VAR\_SALESMAN\_NAME is empty, then: * Display message “Salesman [INPUT Salesman Code] not found!”, [OK] * Set focus to Salesman [To [Code]] field * Else, set Salesman [To [Name]]’s value = VAR\_SALESMAN\_NAME * - |
| Salesman [To [Lookup]] |  | Lookup | * Enter lookup program PML00500 with parameters: * VAR\_COMPANY\_ID * Selected Property Id * VAR\_USER\_ID   Then save the result to VAR\_SALESMAN\_RESULT   * If VAR\_SALESMAN\_RESULT is empty, then: * Display message “Salesman [INPUT Salesman Code] not found!”, [OK] * Set focus to Salesman [To [Code]] field * Else if VAR\_SALESMAN\_RESULT exists, then: * Set Salesman [To [Code]] = CSALESMAN\_ID * Set Salesman [To [Name]] = CSALESMAN\_NAME * - |
| Salesman [To [Name]] |  | Text box | * Disabled * - |
| Period [From [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 [From [Month]] |  | Combo box  Set VAR\_PERIOD\_MONTH\_LIST (as array of object) with VAR\_MONTH\_NUMBER and VAR\_MONTH\_NAME.  **[--CR01]**  Set VAR\_MONTH\_NUMBER = 01, 02, … , 12  Set VAR\_MONTH\_NAME = January, February, … , December  ~~Set VAR\_PERIOD\_FROM\_MONTH\_LIST with~~  ~~EXEC RSP\_GS\_GET\_PERIOD\_DT\_LIST~~  ~~VAR\_COMPANY\_ID~~  ~~, (Period [From][Year])~~  **[--/CR01]**  Set default combo box value month name based on  SELECT MONTH(dbo.RFN\_GET\_DB\_TODAY (VAR\_COMPANY\_ID)) | * Mandatory * - |
| Period [To [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 [To [Month]] |  | Combo box  Set VAR\_PERIOD\_MONTH\_LIST (as array of object) with VAR\_MONTH\_NUMBER and VAR\_MONTH\_NAME.  **[--CR01]**  Set VAR\_MONTH\_NUMBER = 01, 02, … , 12  Set VAR\_MONTH\_NAME = January, February, … , December  ~~Set VAR\_PERIOD\_FROM\_MONTH\_LIST with~~  ~~EXEC RSP\_GS\_GET\_PERIOD\_DT\_LIST~~  ~~VAR\_COMPANY\_ID~~  ~~, (Period [From][Year])~~  **[--/CR01]**  Set default combo box value month name based on  SELECT MONTH(dbo.RFN\_GET\_DB\_TODAY (VAR\_COMPANY\_ID)) | * Mandatory * - |
| Report Type |  | Radio button  **[--CR1]**  Radio Value   * ‘1’ = Summary * ‘2’ = Detail   ~~Radio Value~~   * ~~S = Summary~~ * ~~D = Detail~~   **[--/CR1]**  Set default value to Summary | * Mandatory * - |

### PRINT REPORT PROCESS

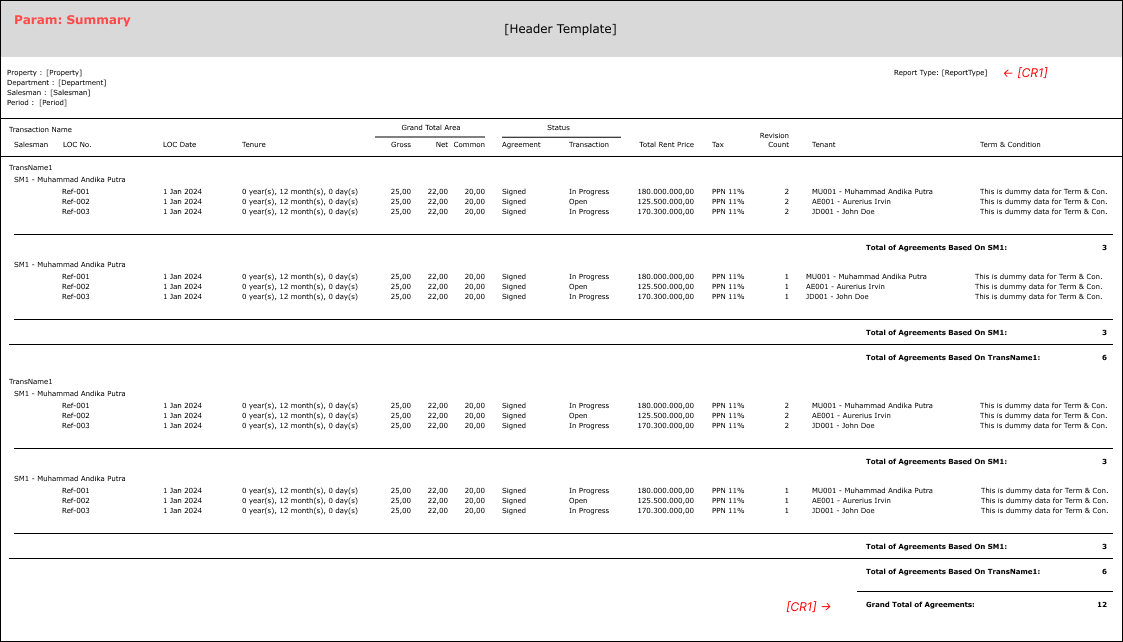
#### REPORT LAYOUT

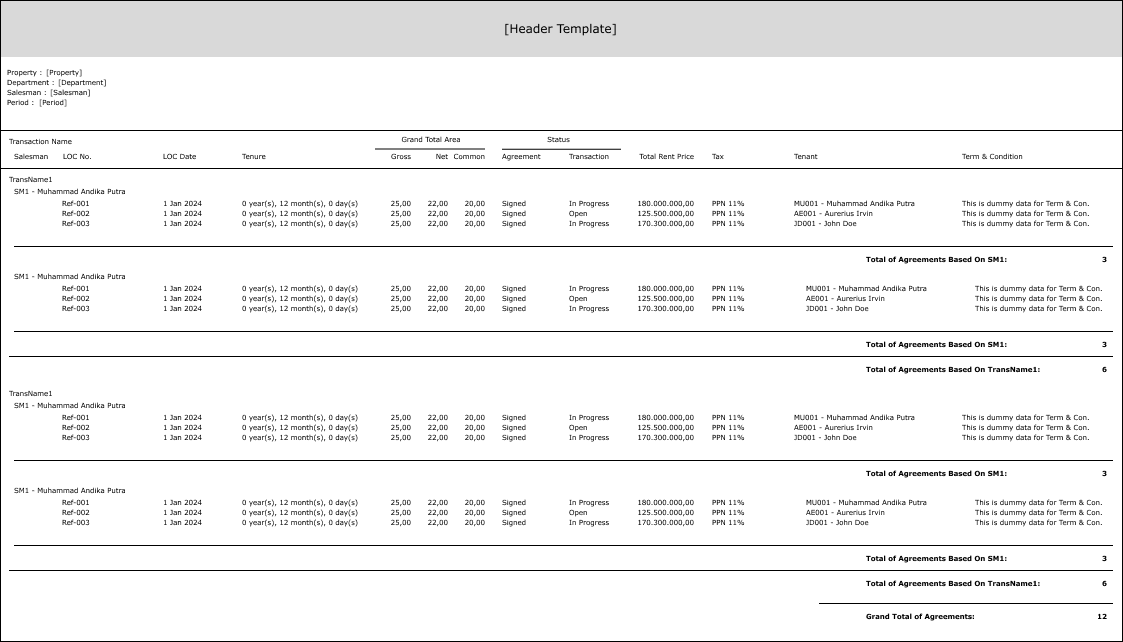
PAPER SIZE: A4 LANDSCAPE

FONT SIZE: 7

REPORT TYPE: SUMMARY

**[--CR1]**





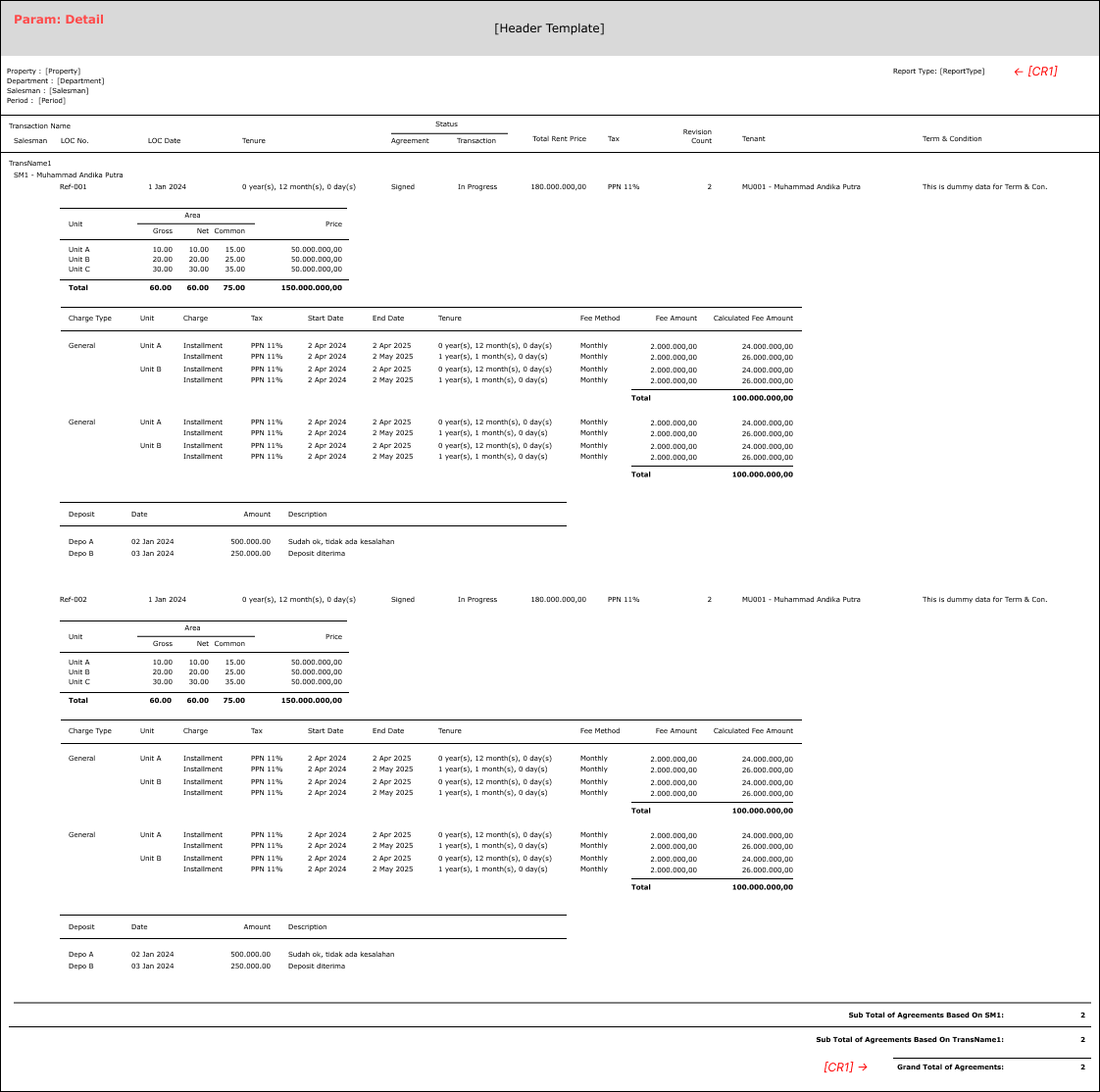
**[--/CR1]**

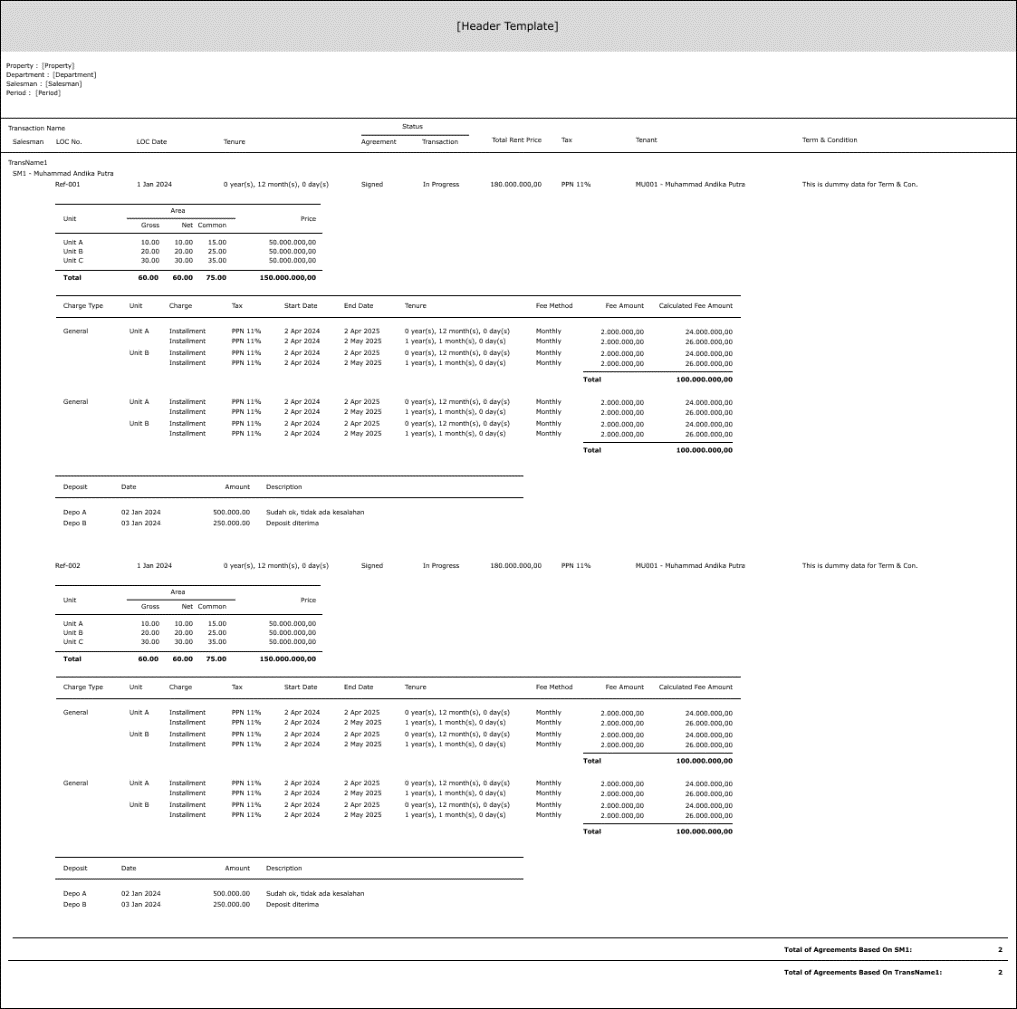
PAPER SIZE: A4 LANDSCAPE

FONT SIZE: 7

REPORT TYPE: DETAIL

**[--CR1]**





**[--/CR1]**

#### REPORT DEFINITION

REPORT SECTION FOR HEADER (IN GENERAL)

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Name** | **Field** | **Validation** |
| HEADER | Property | Selected Property | Show Property ID – Property Name  Ex. “ABCDE – Global Property” |
| Department | Selected Department | IF Department [From [Code]] <> Department [To [Code]], then “Department [From [Name]] (Department [From [Code]]) – Department [To [Name]] (Department [To [Code]])”  ELSE, “Department [From [Name]] (Department [From [Code]])”  Ex. 1  Selected from: MNY – Department Keuangan  Selected to: ITSI – Department Sistem Informasi  Hence, show “Department Keuangan (MNY) – Department Sistem Informasi (ITSI)”  Ex. 2  Selected from: ITSI – Department Sistem Informasi  Selected to: ITSI – Department Sistem Informasi  Hence, show “Department Sistem Informasi (ITSI)” |
| Salesman | Selected Salesman | IF Salesman [From [Code]] <> Salesman [To [Code]], then “Salesman [From [Name]] (Salesman [From [Code]]) – Salesman [To [Name]] (Salesman [To [Code]])”  ELSE, “Salesman [From [Name]] (Salesman [From [Code]])”  Ex. 1  Selected from: MU01 – Andika Putra  Selected to: AE01 – Aurerius Irvin  Hence, show “Andika Putra (MU01) – Aurerius Irvin (AE01)”  Ex. 2  Selected from: MU01 – Andika Putra  Selected to: MU01 – Andika Putra  Hence, show “Andika Putra (MU01)” |
| Period | Selected Period | If Period [From] <> Period [To], then show “MMM yyyy – MMM yyyy”  Ex. Period [From] = “200109” and Period [To] = “200110”, then show “September 2001 – October 2001”  Else, show “MMM yyyy"  Ex. Period [From] = “200109” and Period [From] = “200109”, then show “September 2001” |
| **[--CR1]**  Report Type  **[--/CR1]** | Selected Report Type | Show Report Type’s value  Ex. “Summary” |

REPORT SECTION FOR SUMMARY

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Name** | **Field** | **Validation** |
| REPORT GROUP | Transaction | CTRANS\_CODE |  |
| Salesman | CSALESMAN\_ID |  |
| GROUP BY TRANSACTION [HEADER] | Transaction Name | CTRANS\_NAME |  |
| GROUP BY SALESMAN [HEADER] | Salesman | CSALESMAN\_ID - CSALESMAN\_NAME |  |
| DETAIL | LOC No. | CREF\_NO |  |
| LOC Date | CREF\_DATE |  |
| Tenure | CTENURE |  |
| Grand Total Area (Gross) | NTOTAL\_GROSS\_AREA\_SIZE |  |
| Grand Total Area (Net) | NTOTAL\_NET\_AREA\_SIZE |  |
| Grand Total Area (Common) | NTOTAL\_COMMON\_AREA\_SIZE |  |
| Agreement Status | CAGREEMENT\_STATUS\_NAME |  |
| Transaction Status | CTRANS\_STATUS\_NAME |  |
| Total Rent Price | NTOTAL\_PRICE |  |
| Tax | CTAX |  |
| Tenant | CTENANT\_ID - CTENANT\_NAME |  |
| Term & Condition | CTC\_MESSAGE |  |
| GROUP BY SALESMAN [FOOTER] | Sub Total of Agreement |  | Count of CREF\_NO |
| GROUP BY TRANSACTION [FOOTER] | Sub Total of Agreement |  | Count of CREF\_NO |
| SUMMARY | Grand Total of Agreement |  | Count of CREF\_NO |

REPORT SECTION FOR DETAIL

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Name** | **Field** | **Validation** |
| REPORT GROUP | Transaction | CTRANS\_CODE |  |
| Salesman | CSALESMAN\_ID |  |
| LOC No. | CREF\_NO |  |
| GROUP BY TRANSACTION [HEADER] | Transaction Name | CTRANS\_NAME |  |
| GROUP BY SALESMAN [HEADER] | Salesman | CSALESMAN\_ID - CSALESMAN\_NAME |  |
| GROUP BY LOC NO. [HEADER] | LOC No. | CREF\_NO |  |
| LOC Date | CREF\_DATE |  |
| Tenure | CTENURE |  |
| Agreement Status | CAGREEMENT\_STATUS\_NAME |  |
| Transaction Status | CTRANS\_STATUS\_NAME |  |
| Total Rent Price | NTOTAL\_PRICE |  |
| Tax | CTAX |  |
| Tenant | CTENANT\_ID - CTENANT\_NAME |  |
| Term & Condition | CTC\_MESSAGE |  |
| DATA UNIT [HEADER] | - | - | If (CUNIT\_DETAIL\_ID is null), then suppress the section |
| DATA UNIT [DETAIL] | Unit | CUNIT\_DETAIL\_ID - CUNIT\_DETAIL\_NAME | If (CUNIT\_DETAIL\_ID is null), then suppress the section  For this Section, use a Data Transfer Object, named VAR\_DTO\_UNIT with the following groups:   * CTRANS\_CODE * CSALESMAN\_ID * CREF\_NO * CUNIT\_DETAIL\_ID |
| Gross Area | NUNIT\_DETAIL\_GROSS\_AREA\_SIZE |  |
| Net Area | NUNIT\_DETAIL\_NET\_AREA\_SIZ |  |
| Common Area | NUNIT\_DETAIL\_COMMON\_AREA\_SIZE |  |
| Rent Price | NUNIT\_DETAIL\_PRICE |  |
| DATA UNIT [FOOTER] | Total of Gross Area | **[--CR1]**  ~~NUNIT\_TOTAL\_GROSS\_AREA\_SIZE~~  **[--/CR1]** | If (CUNIT\_DETAIL\_ID is null), then suppress the section  Use Formula sum of DATA UNIT [DETAIL [NUNIT\_DETAIL\_GROSS\_AREA\_SIZE]] |
| Total of Net Area | **[--CR1]**  ~~NUNIT\_TOTAL\_NET\_AREA\_SIZE~~  **[--/CR1]** | Use Formula sum of DATA UNIT [DETAIL [NUNIT\_DETAIL\_NET\_AREA\_SIZE]] |
| Total of Common Area | **[--CR1]**  ~~NUNIT\_TOTAL\_COMMON\_AREA\_SIZE~~  **[--/CR1]** | Use Formula sum of DATA UNIT [DETAIL [NUNIT\_DETAIL\_COMMON\_AREA\_SIZE]] |
| Total of Price | **[--CR1]**  ~~NUNIT\_TOTAL\_PRICE~~  **[--/CR1]** | Use Formula sum of DATA UNIT [DETAIL [NUNIT\_DETAIL\_PRICE]] |
| DATA CHARGE [HEADER] | Charge Type | CCHARGE\_DETAIL\_TYPE\_NAME | If (CCHARGE\_DETAIL\_TYPE\_NAME is null), then suppress the section  For this Section, use a Data Transfer Object, named VAR\_DTO\_CHARGE with the following groups:   * CTRANS\_CODE * CSALESMAN\_ID * CREF\_NO * CCHARGE\_DETAIL\_TYPE\_NAME |
| DATA CHARGE [UNIT [HEADER]] | Unit | CCHARGE\_DETAIL\_UNIT\_NAME | If (CCHARGE\_DETAIL\_TYPE\_NAME is null), then suppress the section  For this Section, use a Data Transfer Object, named VAR\_DTO\_CHARGE\_UNIT with the following groups:   * CTRANS\_CODE * CSALESMAN\_ID * CREF\_NO * CCHARGE\_DETAIL\_TYPE\_NAME * CCHARGE\_DETAIL\_UNIT\_NAME |
| DATA CHARGE [UNIT [DETAIL]] | Charge | CCHARGE\_DETAIL\_CHARGE\_NAME | If (CCHARGE\_DETAIL\_TYPE\_NAME is null), then suppress the section |
| Tax | CCHARGE\_DETAIL\_TAX\_NAME |  |
| Start Date | CCHARGE\_DETAIL\_START\_DATE |  |
| End Date | CCHARGE\_DETAIL\_END\_DATE |  |
| Tenure | CCHARGE\_DETAIL\_TENURE |  |
| Fee Method | CCHARGE\_DETAIL\_FEE\_METHOD |  |
| Fee Amount | NCHARGE\_DETAIL\_FEE\_AMOUNT |  |
| Calculated Fee Amount | NCHARGE\_DETAIL\_CALCULATED\_FEE\_AMOUNT |  |
| DATA CHARGE [UNIT [FOOTER]] | - | - | If (CCHARGE\_DETAIL\_TYPE\_NAME is null), then suppress the section |
| DATA CHARGE [FOOTER] | Total of Calculated Fee Amount | **[--CR1]**  ~~NCHARGE\_DETAIL\_SUBTOTAL\_CALCULATED\_FEE\_AMOUNT~~  **[--/CR1]** | If (CCHARGE\_DETAIL\_TYPE\_NAME is null), then suppress the section  Use Formula sum of DATA CHARGE [HEADER [NCHARGE\_DETAIL\_CALCULATED\_FEE\_AMOUNT]] |
| DATA DEPOSIT [HEADER] | - | - | If (CDEPOSIT\_DETAIL\_ID is null), then suppress the section |
| DATA DEPOSIT [DETAIL] | Deposit | CDEPOSIT\_DETAIL\_ID | If (CDEPOSIT\_DETAIL\_ID is null), then suppress the section |
| Date | CDEPOSIT\_DETAIL\_DATE |  |
| Amount | NDEPOSIT\_DETAIL\_AMOUNT |  |
| Description | CDEPOSIT\_DETAIL\_DESCRIPTION |  |
| DATA DEPOSIT [FOOTER] | - | - | If (CDEPOSIT\_DETAIL\_ID is null), then suppress the section |
| GROUP BY LOC NO. [FOOTER] | - | - | If (CDEPOSIT\_DETAIL\_ID is null), then suppress the section |
| GROUP BY SALESMAN [FOOTER] | Sub Total of Agreement |  | Count of CREF\_NO |
| GROUP BY TRANSACTION [FOOTER] | Sub Total of Agreement |  | Count of CREF\_NO |
| FOOTER | Grand Total of Agreement |  | Count of CREF\_NO |

#### VALIDATION

|  |  |  |
| --- | --- | --- |
| **Field** | **Condition** | **Validation Message** |
| Property | Empty | **[--CR1]**  Please select Property  ~~Please select Property!~~  **[--/CR1]** |
| Department | Department [From [Code]] is empty **or** Department [To [Code]] is empty | **[--CR1]**  Please select Department  ~~Please select Department (From)!~~  ~~Please select Department (To)!~~  ~~Department (To) is higher than Department (From)~~  **[--/CR1]** |
| Salesman | Salesman [From [Code]] is empty **or** Salesman [To [Code]] is empty | **[--CR1]**  Please select Salesman  ~~Please select Salesman (From)!~~  ~~Please select Salesman (To)!~~  ~~Salesman (To) is higher than Salesman (From)~~  **[--/CR1]** |
| Period | Period [From] is higher than Period [To] | **[--CR1]**  Period (From) cannot be higher than Period (To)  ~~Period (To) is higher than Period (From)!~~  **[--/CR1]** |

#### ACTION

|  |
| --- |
|  |
| * Execute this stored procedure:   RSP\_PMR00150\_GET\_REPORT  VAR\_COMPANY\_ID  , selected Property ID  , selected Department [From [Code]]  , selected Department [To [Code]]  , selected Salesman [From [Code]]  , selected Salesman [To [Code]]  , selected Period [From]  , selected Period [To]  **[--CR1]**  , selected Report Type  **[--/CR1]**  , VAR\_LANGUAGE\_ID  **[--CR1]**  Note for sending those parameters to execute the stored procedure above:   * Selected Period [From]’s value should be “yyyymm” * Selected Period [To]’s value should be “yyyymm” * Selected Report Type’s value should be either ‘1’ or ‘2’ * **!IMPORTANT!: DIFFERENT SELECTED REPORT TYPE RESULTING IN DIFFERENT OUTPUT**   ~~There are two types of report;~~  ~~If type = S, then open report for Summary~~  ~~Else if type = D, then open report for Detail~~  **[--/CR1]**   * Ex. 1 for Summary   EXEC RSP\_PMR00150\_GET\_REPORT  'RCD'  , 'JBMPC'  , 'HRD'  , 'HRD'  , 'Alvan'  , 'Alvan'  , '202401'  , '202412'  , '1'  , 'EN'   * Ex. 2 for Detail   EXEC RSP\_PMR00150\_GET\_REPORT  'RCD'  , 'JBMPC'  , 'HRD'  , 'HRD'  , 'Alvan'  , 'Alvan'  , '202401'  , '202412'  , '2'  , 'EN'   * - |

# CR LIST

|  |  |  |
| --- | --- | --- |
| **CR** | **Description** | **Notes** |
| [CR1] | * Renamed program name, from “LOC Status” → “LOC List” * Changed the method on getting the month. Previously, it was taken from Stored Procedure VAR\_PERIOD\_FROM\_MONTH\_LIST, but it does not contain month name in return * Changed the value of Report Type value. Previously, its value was ‘S’ or ‘D’ * Added Report Type parameter in every report (Summary & Detail) header * Deleted five variables on Report (Param: Detail), which are: * NUNIT\_TOTAL\_GROSS\_AREA\_SIZE * NUNIT\_TOTAL\_NET\_AREA\_SIZE * NUNIT\_TOTAL\_COMMON\_AREA\_SIZE * NUNIT\_TOTAL\_PRICE * NCHARGE\_DETAIL\_SUBTOTAL\_CALCULATED\_FEE\_AMOUNT   Instead, those values above can be received through the calculation in Middle-End   * Changed the validation message on every condition | 28 June 2024  By MU |

#### TEMPLATE

#### VALIDATION

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

#### ACTION

|  |
| --- |
|  |
|  |