LAMPIRAN

Lampiran 1: Program *Objective Function*

```
function [Cost, nPenalty, iBCCDesign, TableList] = obj_func3(G,x)
% Main objective function
% Cost = obj_func3(G,x)
% Where:
% G is the global structure
% x is an array consisting of beam and column dimensions
% [T2Beam, T3Beam, T2Column, T3Column]
d1 = datetime;
%% Get number of beams and columns from global
nBeams = G.Constants.nBeams;
nColumns = G.Constants.nColumns;
%% Get price
conc_strength = "fc " +
G.Tables.MaterialPropertiesConcreteData.Fc/1000;
% conc_strength = "fc 40"; % User input
conc_price = G.PriceList.Harga(find(G.PriceList.("Material") ==
conc_strength));
steel_price = G.PriceList.Harga(find(G.PriceList.("Material") ==
"Baja"));
%% Separate x into beams and columns
% Rounds up each dimension to the nearest 50 cm
x = round(x./0.05).*0.05;
T2Beam = round(x(1:nBeams)./0.05)*0.05;
T3Beam = round(x((nBeams+1):(nBeams*2))./0.05)*0.05;
T2Column = round(x((nBeams*2+1):(nBeams*2+nColumns))./0.05)*0.05;
T3Column =
round(x((nBeams*2+nColumns+1):(nBeams*2+nColumns*2))./0.05)*0.05;
%% Dimensions constraint penalty
penalty1 = constraint_dimension(T2Beam,T3Beam,T2Column,T3Column);
if penalty1 > 0 %returns a large penalty if fails
Cost = penalty1*10^15;
nPenalty = penalty1*25;
Cost = Cost+nPenalty/1000;
return
end

%% Reset reinforcement
```

Lampiran 1: Program *Objective Function* (Lanjutan)

```
retInitialize = resetSections2(G,x);
if retInitialize == 1 % returns function if fails
Cost = 0; nPenalty = Inf;
return
end
%% Run the structural analysis
retAnalysis = G.API.Analyze.RunAnalysis;
%% Scale response spectrum base shear // add torsion here.
RS_ScaleSF(G);
RS_EccScale(G);
%% Run concrete design on special moment beams and top columns only.
retDeselectAll=G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemTyp
e.Group;
retSelectBeams=G.API.FrameObj.SetSelected("02_SMBEAMS",true,ETABSv1.eI
temType.Group;
%Data might be corrupted/be inconsistent if done this way
retStartDesign = G.API.DesignConcrete.StartDesign;
BeamRebarTable = beamRebarDesign1a(G);
%% Reinput reinforcement to ETABS.
retSetRebar1 = setRebar1_1(G,BeamRebarTable);
%% Rerun analysis and concrete design.
retAnalysis = G.API.Analyze.RunAnalysis;
%% Column design.
% Top column design
tic
retChangeBCCPreference = G.API.ACI318_14.SetPreference(4,1); % turn
off design for BCC
retSelectFrames=G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemTy
pe.Group);
retSelectBeams=G.API.FrameObj.SetSelected("02_SMBEAMS",true,ETABSv1.eI
temType.Group);
retDeselectTopColumns=G.API.FrameObj.SetSelected("04_TOPCOLUMNS",true,
ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
TopColumnReinTable = columnRebarDesign1a(G);
% All column design
retChangeBCCPreference = G.API.ACI318_14.SetPreference(4,2); % turn on
design for BCC
```

Lampiran 1: Program *Objective Function* (Lanjutan)

```
retSelectFrames =
G.API.FrameObj.SetSelected("All",true,ETABSv1.eItemType.Group);
retDeselectTopColumns=G.API.FrameObj.SetSelected("04_TOPCOLUMNS",false
,ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
ColumnRebarTable = columnRebarDesign1b(G,TopColumnReinTable);
retSetRebar2 = setRebar2_1(G,ColumnRebarTable);
toc
%% Set rebar 3 column design
iBCCDesign = 0;
RedesignFlag = true;
maxiBCCDesign = 1;
while RedesignFlag == true && iBCCDesign <= maxiBCCDesign
iBCCDesign = iBCCDesign + 1;
retAnalysis = G.API.Analyze.RunAnalysis;
retDeselectAll =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
[~, RedesignFlag] = columnAsDesign(G);
end
for iii=1:2 %do this twice
retAnalysis = G.API.Analyze.RunAnalysis;
retDeselectAll =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
%Column shear design
ColumnTable = columnAvDesign(G);
setRebar3_1(G,ColumnTable);
end
% Final concrete design
retAnalysis = G.API.Analyze.RunAnalysis;
retDeselectAll =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
%% Check for constraints
%Beam shear design
[BeamRebarTable, nBeamPenalty] = beamRebarDesign1b(G,BeamRebarTable)
nColumnPenalty = constraint_column(G)
```

Lampiran 1: Program *Objective Function* (Lanjutan)

```
nJointPenalty = constraint_BCC_JS(G)
nDriftPenalty = constraint_drift(G)
%% Calculate steel cost
%% Reinforcement tables
FrameSum = G.Tables.FrameSum;
FrameSum = FrameSum(:,["Analysis Section","Length"]);
FrameSum = renamevars(FrameSum,'Analysis Section','Name');
BeamRebarTable = innerjoin(BeamRebarTable,FrameSum);
StationLength = [0.33, 0.34, 0.33, 0.33, 0.34, 0.33];
BeamTotalRebar =
sum(BeamRebarTable.("AsProv").*StationLength.*BeamRebarTable.("Length"
),[1 2]);
ColumnTable = innerjoin(ColumnTable,FrameSum);
ColumnTotalRebar = sum(ColumnTable.("As").*ColumnTable.("Length"));
Trv1(:,1) =
ceil(0.33.*BeamRebarTable.("LnBeam")./BeamRebarTable.TrvSpacing(:,1));
Trv1(:,2) =
ceil(0.34.*BeamRebarTable.("LnBeam")./BeamRebarTable.TrvSpacing(:,2));
Trv1(:,3) =
ceil(0.33.*BeamRebarTable.("LnBeam")./BeamRebarTable.TrvSpacing(:,3));
Trv1(isnan(Trv1))=0;
Trv1(:,4) = (BeamRebarTable.("Width")-
0.04*2)*2+((BeamRebarTable.("Depth")-
0.04*2).*BeamRebarTable.nTrvLegs(:,1));
Trv1(:,5) = (BeamRebarTable.("Width")-
0.04*2)*2+((BeamRebarTable.("Depth")-
0.04*2).*BeamRebarTable.nTrvLegs(:,2));
Trv1(:,6) = (BeamRebarTable.("Width")-
0.04*2)*2+((BeamRebarTable.("Depth")-
0.04*2).*BeamRebarTable.nTrvLegs(:,3));
Trv1(:,7) = Trv1(:,1).*Trv1(:,4);
Trv1(:,8) = Trv1(:,2).*Trv1(:,5);
Trv1(:,9) = Trv1(:,3).*Trv1(:,6);
Trv1(:,10) = Trv1(:,7)+Trv1(:,8)+Trv1(:,9);
Trv1(:,11) =
Trv1(:,10).*(0.25.*3.14./1000000.*BeamRebarTable.("dtMinimum").^2);
ShearBeam = sum(Trv1(:,11));
Trv2(:,1) = ceil(ColumnTable.("Length")./ColumnTable.("SpacingUse"));
```

Lampiran 1: Program *Objective Function* (Lanjutan)

```
Trv2(:,2) = ColumnTable.("Depth")-0.04*2;
Trv2(:,3) = ColumnTable.("Width")-0.04*2;
Trv2(:,4) =
((Trv2(:,2).*ColumnTable.("NumberOfLegsMaj")+Trv2(:,3).*ColumnTable.("
NumberOfLegsMin"))).*Trv2(:,1);
Trv2(:,5) =
Trv2(:,4).*0.25.*3.14.*(ColumnTable.("DiameterTransversal")./1000).^2;
ShearColumn = sum(Trv2(:,5));
TotalRebar = ColumnTotalRebar + BeamTotalRebar + ShearBeam +
ShearColumn;
CostConcrete = smf_weight(G)./23.56.*conc_price;
CostSteel = TotalRebar * 7850 * steel_price;
Cost = CostConcrete + CostSteel;
nPenalty = nBeamPenalty + nColumnPenalty + nJointPenalty +
nDriftPenalty;
Cost = round(Cost,0) + nPenalty/1000;
TableList.BeamPenalty = nBeamPenalty;
TableList.ColumnPenalty = nColumnPenalty;
TableList.JointPenalty = nJointPenalty;
TableList.DriftPenalty = nDriftPenalty;
d2 = datetime;
time = d2-d1;
disp(time)
```

Lampiran 2: Algoritma Optimasi SOS

```
clear;clc;close all;
tic;
% Connect MATLAB to ETABS
parentFolder = fileparts(pwd);
listing = dir(parentFolder);
for iFiles = 1:length(listing)
if listing(iFiles).isdir
addpath(parentFolder + "\" + listing(iFiles).name)
end
end
G = globalVar;
%% Added ETABSObject to all fobj
d1=datetime;
initial_population = readmatrix("Population40.csv");
initial_sections = initial_population(1,:);
% editable
lb_1=getDimLowerBound(G.API.DatabaseTables);
lb = lb_1;
ub_1=ones(1,length(lb))*1.5; % highly consider changing this
ub = ub_1; % unconstrainted upper bounds
fobj=@obj_func3;
iteration=30;
ecosize=height(initial_population);
n=length(lb);
maxFE=iteration*ecosize*4; % multiply by 4
FE=0;
fitnessNew1=zeros(ecosize,2);
ecoNew1=zeros(ecosize,n);
fitnessNew2=zeros(ecosize,2);
ecoNew2=zeros(ecosize,n);
fitnessNew3=zeros(ecosize,2);
ecoNew3=zeros(ecosize,n);
fitnessNew4=zeros(ecosize,2);
ecoNew4=zeros(ecosize,n);
fitness = zeros(ecosize,2);
fitnessmatrix = zeros(ecosize,iteration+1);
fitnessmatrix(:,1) = fitness(:,1);
iter=0;
```

Lampiran 2: Algoritma Optimasi SOS (Lanjutan)

```
% -- Initial population;
eco = initial_population;
ecomatrix = zeros([size(eco), iteration+1]);
%% Ecosystem Initialization
for i=1:ecosize
% Initialize the organisms randomly in the ecosystem
% Evaluate the fitness of the new solution
[fitness(i,1),fitness(i,2)]=fobj(G,eco(i,:));
end
fmax=max(fitness(:,1).*(fitness(:,2)==0));
for i=1:ecosize
if fitness(i,2)>0
fitness(i,1)=fmax+fitness(i,2);
end
end
% sort & update the best initial organism
[fitness, l]=sortrows(fitness,1); eco=eco(l,:);
[fitness, m]=sortrows(fitness,2); eco=eco(m,:);
[bestFitness]=fitness(1,1); bestOrganism=eco(1,:);
ii=1;
%% Main Looping
while FE<maxFE
for i=1:ecosize % Organisms' Looping
% Update the best Organism
[~,idx]=min(fitness(:,1)); bestOrganism=eco(idx,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mutualism Phase
% Choose organism j randomly other than organism i
j=i;
while i==j
seed=randperm(ecosize);
j=seed(1);
end
% Determine Mutual Vector & Beneficial Factor
mutualVector=mean([eco(i,:); eco(j,:)]);
BF1=round(1+rand); BF2=round(1+rand);
% Calculate new solution after Mutualism Phase
ecoNew1(i,:)=eco(i,:)+rand(1,n).*(bestOrganism-BF1.*mutualVector);
```

Lampiran 2: Algoritma Optimasi SOS (Lanjutan)

```
ecoNew2(j,:)=eco(j,:)+rand(1,n).*(bestOrganism-BF2.*mutualVector);
ecoNew1(i,:)=bound(ecoNew1(i,:),ub,lb);
ecoNew2(j,:)=bound(ecoNew2(j,:),ub,lb);
% Evaluate the fitness of the new solution
[fitnessNew1(i,1),fitnessNew1(i,2)]=fobj(G,ecoNew1(i,:));
[fitnessNew2(j,1),fitnessNew2(j,2)]=fobj(G,ecoNew2(j,:));
if fitnessNew1(i,2)==0
elseif fitnessNew1(i,2)>0
fitnessNew1(i,1)=fmax+fitnessNew1(i,2);
end
if fitnessNew2(j,2)==0
elseif fitnessNew2(j,2)>0
fitnessNew2(j,1)=fmax+fitnessNew2(j,2);
end
if fitnessNew1(i,2)==fitness(i,2)
if fitnessNew1(i,1)<fitness(i,1)
fitness(i,:)=fitnessNew1(i,:);
eco(i,:)=ecoNew1(i,:);
end
elseif fitnessNew1(i,2)<fitness(i,2)
fitness(i,:)=fitnessNew1(i,:);
eco(i,:)=ecoNew1(i,:);
end
if fitnessNew2(j,2)==fitness(j,2)
if fitnessNew2(j,1)<fitness(j,1)
fitness(j,:)=fitnessNew2(j,:);
eco(j,:)=ecoNew2(j,:);
end
elseif fitnessNew2(j,2)<fitness(j,2)
fitness(j,:)=fitnessNew2(j,:);
eco(j,:)=ecoNew2(j,:);
end
% Accept the new solution if the fitness is better
% Increase the number of function evaluation counter
FE=FE+2;
% End of Mutualism Phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Lampiran 2: Algoritma Optimasi SOS (Lanjutan)

```
 % Commensialism Phase
 % Choose organism j randomly other than organism i
j=i;
while i==j
seed=randperm(ecosize);
j=seed(1);
end
% Calculate new solution after Commensalism Phase
ecoNew3(i,:)=eco(i,:)+(rand(1,n)*2-1).*(bestOrganism-eco(j,:));
ecoNew3(i,:)=bound(ecoNew3(i,:),ub,lb);
% Evaluate the fitness of the new solution
[fitnessNew3(i,1),fitnessNew3(i,2)]=fobj(G,ecoNew3(i,:));
if fitnessNew3(i,2)==0
elseif fitnessNew3(i,2)>0
fitnessNew3(i,1)=fmax+fitnessNew3(i,2);
end
if fitnessNew3(i,2)==fitness(i,2)
if fitnessNew3(i,1)<fitness(i,1)
fitness(i,:)=fitnessNew3(i,:);
eco(i,:)=ecoNew3(i,:);
end
elseif fitnessNew3(i,2)<fitness(i,2)
fitness(i,:)=fitnessNew3(i,:);
eco(i,:)=ecoNew3(i,:);
end
% Increase the number of function evaluation counter
FE=FE+1;
% End of Commensalism Phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parasitism Phase
% Choose organism j randomly other than organism i
j=i;
while i==j
seed=randperm(ecosize);
j=seed(1);
end

% Determine Parasite Vector & Calculate the fitness
```

Lampiran 2: Algoritma Optimasi SOS (Lanjutan)

```
parasiteVector=eco(i,:);
seed=randperm(n);
pick=seed(1:ceil(rand*n));  % select random dimension
parasiteVector(:,pick)=rand.*(ub(pick)-lb(pick))+lb(pick);
ecoNew4(i,:)=parasiteVector;
% Evaluate the fitness of the new solution
[fitnessNew4(i,1),fitnessNew4(i,2)]=fobj(G,ecoNew4(i,:));
if fitnessNew4(i,2)==0
elseif fitnessNew4(i,2)>0
fitnessNew4(i,1)=fmax+fitnessNew4(i,2);
end
% Kill organism j and replace it with the parasite
% if the fitness is lower than the parasite
if fitnessNew4(i,2)==fitness(j,2)
if fitnessNew4(i,1)<fitness(j,1)
fitness(j,:)=fitnessNew4(i,:);
eco(j,:)=ecoNew4(i,:);
end
elseif fitnessNew4(i,2)<fitness(j,2)
fitness(j,:)=fitnessNew4(i,:);
eco(j,:)=ecoNew4(i,:);
end
% Increase the number of function evaluation counter
FE=FE+1;
% End of Parasitism Phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end % End of Organisms' Looping
fmax=max(fitness(:,1).*(fitness(:,2)==0));
for i=1:ecosize
if fitness(i,2)>0
fitness(i,1)=fmax+fitness(i,2);
end
end
% sort and update the best organism
[fitness, l]=sortrows(fitness,1);eco=eco(l,:);
[fitness, m]=sortrows(fitness,2); eco=eco(m,:);
bestOrganism=eco(1,:); bestFitness=fitness(1,1);
```

Lampiran 2: Algoritma Optimasi SOS (Lanjutan)

```
fitnessmatrix(:,ii) = fitness(:,1);
ecomatrix(:,:,ii) = eco;
ii=ii+1;
end % End of Main Looping
disp(['Funnum: ', num2str(funnum)])
disp(['fitness: ', num2str(bestFitness)])
disp(['Solution: ', num2str(bestOrganism)])
%% Post-processing
d2 = datetime;
runtime = d2-d1; disp(runtime);
writematrix([fitness,eco],"Results -
"+string(datetime('now','Format','yyyy-MM-dd_HH_mm_ss'))+".csv")
casefilename='CaseSOS2-'+string(datetime('now','Format','yyyy-MM-
dd_HH_mm_ss'))+'.mat';
save(casefilename)
% Check the boundary limit
function a=bound(a,ub,lb)
a(a>ub)=ub(a>ub); a(a<lb)=lb(a<lb);
end
function fitness=fitnessEval(fitness,eco,funnum,fmax,data)
if fitness(1,2)==0
fitness(1,1)=fobj(eco,funnum,data);
elseif fitness(1,2)>0
fitness(1,1)=fmax+fitness(1,2);
end
end
function
[fitnesseco]=compareTwoSolution(fitnessNew,fitness,ecoNew,eco)
if fitnessNew(1,2)==fitness(1,2)
if fitnessNew(1,1)<fitness(1,1)
fitness=fitnessNew;
eco=ecoNew;
end
elseif fitnessNew(1,2)<fitness(1,2)
fitness=fitnessNew;
eco=ecoNew;
end
end
```

Lampiran 3: globalVar

```
G = globalVar()
% This function initializes global variables, constaints, and API
objects
% and stores it in the struct G.
%%Longitudinal and transverse reinforcement considered - user input
BeamLongitudinalBars = [19, 22, 25];
BeamTransversalBars = [10];
ColumnLongitudinalBars = [22, 25];
ColumnTransversalBars = [10, 13];
%%Concrete cover and minimum spacing requirements
ConcreteCover = 0.04;
MinimumSpacingTrv = 0.05;
MinimumSpacingParallel = 0.025;
MaximumLayers = 7;
MultipleSpacingTrv = 0.025;
%%Material strength
fcBeam = 40;
fcColumn = 40;
fyt = 420;
fy = 420;
%%Constraint handling - userinput
%%Column constraints
MaxPMMRatio = 1.025; % 2.5% tolerance
%%Joint constraints
MaxBCCRatio = 1.025; %2.5% tolerance
MaxJSRatio = 1;
%%Drift constraints - user input
Sds = 0.6; %Sds used in designed
Ie = 1.5; % 11.5.1
Cd = 5.5; % Table 12.2-1
MaxDrift = 0.01; % 12.12.1
%% API and Program File Location
ProgramPath = "C:\Program Files\Computers and Structures\ETABS
21\ETABS.exe";
APIDLLPath = 'C:\Program Files\Computers and Structures\ETABS
21\ETABSv1.dll';
if ~isfile(ProgramPath)
```

Lampiran 3: globalVar (Lanjutan)

```
ProgramPath = "C:\Program Files\Computers and Structures\ETABS
21\ETABS.exe";
APIDLLPath = 'C:\Program Files\Computers and Structures\ETABS
21\ETABSv1.dll';
end
%% Attach to API
%%create API helper object
a = NET.addAssembly(APIDLLPath);
helper = ETABSv1.Helper;
helper = NET.explicitCast(helper,'ETABSv1.cHelper');
ETABSObject = helper.GetObject('CSI.ETABS.API.ETABSObject');
ETABSObject = NET.explicitCast(ETABSObject,'ETABSv1.cOAPI');
helper = 0;
%%Access SapModel interface
SapModel = NET.explicitCast(ETABSObject.SapModel,'ETABSv1.cSapModel');
%%Access DatabaseTables interface
DatabaseTables=NET.explicitCast(SapModel.DatabaseTables,'ETABSv1.cData
baseTables');
%%Access DesignConcrete interface
DesignConcrete=NET.explicitCast(SapModel.DesignConcrete,'ETABSv1.cDesi
gnConcrete');
%%Access Analyze interface
Analyze = NET.explicitCast(SapModel.Analyze,'ETABSv1.cAnalyze');
%%Access FrameObj interface
FrameObj = NET.explicitCast(SapModel.FrameObj,'ETABSv1.cFrameObj');
%%Access PropFrame Interface
PropFrame = NET.explicitCast(SapModel.PropFrame,'ETABSv1.cPropFrame');
%%Access ACI318-14 interface
ACI318_14 =
NET.explicitCast(DesignConcrete.ACI318_14,'ETABSv1.cDCoACI318_14');
%%Access LoadCases and ResponseSpectrum interfaces
LoadCases = NET.explicitCast(SapModel.LoadCases,'ETABSv1.cLoadCases');
ResponseSpectrum =
NET.explicitCast(LoadCases.ResponseSpectrum,'ETABSv1.cCaseResponseSpec
trum');
%%Sets units to kN/m/C
ret = SapModel.SetPresentUnits(ETABSv1.eUnits.kN_m_C);
%%Get concrete design code
```

Lampiran 3: globalVar (Lanjutan)

```
CodeName = "";
[ret, CodeName] = DesignConcrete.GetCode(CodeName); %get concrete
design code
% Get Frame Assignments - Summary table for lengths of frame sections
TableKey = "Frame Section Property Definitions - Summary";
FrameSectionPropertyDefinitions = getTableDisplay(DatabaseTables,
TableKey);
TableKey = "Material Properties - Concrete Data";
MaterialPropertiesConcreteData = getTableDisplay(DatabaseTables,
TableKey);
TableKey = "Frame Assignments - Summary";
FrameSum = getTableDisplay(DatabaseTables, TableKey, "01_SMF");
AnalysisSectionVector = FrameSum.("Analysis Section");
nBeams = sum(startsWith(AnalysisSectionVector,"G"));
nColumns = sum(startsWith(AnalysisSectionVector,"K"));
% Remake tables
FrameSum = FrameSum(:,["Story", "Label", "UniqueName", "Length",
"Analysis Section", "Design Section"]);
FrameSectionPropertyDefinitions =
FrameSectionPropertyDefinitions(:,["Name", "Material"]);
MaterialPropertiesConcreteData =
MaterialPropertiesConcreteData(:,["Material","Fc"]);
FrameSectionPropertyDefinitions =
innerjoin(FrameSectionPropertyDefinitions,MaterialPropertiesConcreteDa
ta);
FrameSectionPropertyDefinitions =
renamevars(FrameSectionPropertyDefinitions,"Name","Analysis Section");
FrameSum = innerjoin(FrameSum,FrameSectionPropertyDefinitions);
% Assign to global struct
G.API.ETABSObject = ETABSObject;
G.API.SapModel = SapModel;
G.API.DatabaseTables = DatabaseTables;
G.API.Analyze = Analyze;
G.API.DesignConcrete = DesignConcrete;
G.API.FrameObj = FrameObj;
G.API.ACI318_14 = ACI318_14;
G.API.PropFrame = PropFrame;
G.API.LoadCases = LoadCases;
```

Lampiran 3: globalVar (Lanjutan)

```
G.API.ResponseSpectrum = ResponseSpectrum;
G.Constants.DesignCode = string(CodeName);
G.PriceList = readtable("PriceList.csv");
G.Tables.FrameSum = FrameSum;
G.Tables.FrameSectionPropertyDefinitions =
FrameSectionPropertyDefinitions;
G.Tables.MaterialPropertiesConcreteData =
MaterialPropertiesConcreteData;
G.Constants.Ie = Ie;
G.Constants.Cd = Cd;
G.Constants.maxDrift = MaxDrift;
G.Constants.BeamLongitudinalBars = BeamLongitudinalBars;
G.Constants.BeamTransversalBars = BeamTransversalBars;
G.Constants.ColumnLongitudinalBars = ColumnLongitudinalBars;
G.Constants.ColumnTransversalBars = ColumnTransversalBars;
G.Constants.MaxBCCRatio = MaxBCCRatio;
G.Constants.MaxJSRatio = MaxJSRatio;
G.Constants.ConcreteCover = ConcreteCover;
G.Constants.MinimumSpacingTrv = MinimumSpacingTrv;
G.Constants.MinimumSpacingParallel = MinimumSpacingParallel;
G.Constants.nBeams = nBeams;
G.Constants.nColumns = nColumns;
G.Constants.fcBeam = fcBeam;
G.Constants.fcColumn = fcColumn;
G.Constants.Sds = Sds;
G.Constants.MaximumLayers = MaximumLayers;
G.Constants.fyt = fyt;
G.Constants.fy = fy;
G.Constants.MultipleSpacingTrv = MultipleSpacingTrv;
G.Constants.MaxPMMRatio = MaxPMMRatio;
```

Lampiran 4: InitialPopulationSearch_3

```
%InitialPopulationSearch w/ dimension penalty only
parentFolder = fileparts(pwd);
listing = dir(parentFolder);
for iFiles = 1:length(listing)
if listing(iFiles).isdir
addpath(parentFolder + "\" + listing(iFiles).name)
end
end
G = globalVar;
nBeams = G.Constants.nBeams; nColumns = G.Constants.nColumns;
%%%
nPopulation = 40;
%%%
nEle = nBeams + nColumns;
nVar = nEle*2;
population = zeros(nPopulation,nVar);
fitness = zeros(nPopulation,1);
T2 = zeros(nPopulation, nVar./2);
iSuccess = 1;
initial_sections = readmatrix("InitialSections.csv");
lb_1=getDimLowerBound(G.API.DatabaseTables);
lb_2=initial_sections-0.1;
lb=max(lb_1,lb_2);
ub_1=ones(1,length(lb))*1.5; % highly consider changing this
ub_2=initial_sections+0.2;
ub=min(ub_1,ub_2);
T2ub = ub([1:nBeams,nEle+(1:nColumns)]);
T2lb = lb([1:nBeams,nEle+(1:nColumns)]);
f_obj = @constraint_dimension2;
nFail = 0;
nPenalized = 0;
while iSuccess <= nPopulation
T2(iSuccess,:)=rand(1,nEle).*(T2ub-T2lb)+T2lb;
T2(iSuccess,:)=round(T2(iSuccess,:)./0.05).*0.05;
T2(iSuccess,:)=bound(T2(iSuccess,:),T2ub,T2lb);
population(iSuccess,:)=constrained_dimension3(G,T2(iSuccess,:),lb,ub);
population(iSuccess,:)=bound(population(iSuccess,:),ub,lb);
if iSuccess == 1
```

Lampiran 4: InitialPopulationSearch_3 (Lanjutan)

```
population(1,:) = initial_sections;
end
fitness(iSuccess,1) = f_obj(G,population(iSuccess,:));
if fitness(iSuccess,1) > 0
nFail = nFail + 1;
continue
elseif mod(fitness(iSuccess,1),1) > 0
nPenalized = nPenalized +1;
end
iSuccess = iSuccess + 1;
end
casefilename='Population-'+string(datetime('now','Format','yyyy-MM-
dd_HH_mm_ss'))+'.mat';
```

Lampiran 5: constraint_dimension2

```
function [total_dim_penalty, beam_dim_penalty, column_dim_penalty] =
constraint_dimension2(G, x)
% This function checks the dimension constraints of SMF beams and
columns.
nBeams = G.Constants.nBeams;
nColumns = G.Constants.nColumns;
T2Beam = round(x(1:nBeams)./0.05)*0.05;
T3Beam = round(x((nBeams+1):(nBeams*2))./0.05)*0.05;
T2Column = round(x((nBeams*2+1):(nBeams*2+nColumns))./0.05)*0.05;
T3Column=round(x((nBeams*2+nColumns+1):(nBeams*2+nColumns*2))./0.05)*0
.05;
beam_dim_penalty = sum(T3Beam < 0.3.*T2Beam);
column_dim_penalty = sum([T2Column./T3Column < 0.4, T3Column./T2Column
< 0.4]);
total_dim_penalty = beam_dim_penalty+column_dim_penalty;
if total_dim_penalty > 0
disp("Dimension penalty")
end
```

Lampiran 6: RS_ScaleSF

```
function RS_ScaleSF(G)
%% Scale response spectrum base shear to equivalent lateral force base
shear
tic
ScaleFactor = 0.85;
%% Insert names of load cases here (user input)
% Program assumes the use of Auto Seismic Load to compare with ELF
ELFXName = 'EQX';
ELFYName = 'EQY';
RSXName = 'GEMPAX';
RSYName = 'GEMPAY';
%% Interface access
string_array = NET.createArray('System.String',2);
doublenumber = 1.1;
integer = int32(1);
%%Access SapModel Interface
SapModel = G.API.SapModel;
%%Access Analyze, AnalysisResults, AnalysisResultsSetup interface
Analyze = G.API.Analyze;
Results=NET.explicitCast(SapModel.Results,'ETABSv1.cAnalysisResults');
Setup=NET.explicitCast(Results.Setup,'ETABSv1.cAnalysisResultsSetup');
%%Access ResponseSpectrum interface
LoadCases = NET.explicitCast(SapModel.LoadCases,'ETABSv1.cLoadCases');
ResponseSpectrum=NET.explicitCast(LoadCases.ResponseSpectrum,'ETABSv1.
cCaseResponseSpectrum');
%% Runs analysis
retAnalysis = Analyze.RunAnalysis();
%% Gets Base Shear from Auto ELF and current RS
%%Assigns ELF LoadCase for Output
ret = Setup.DeselectAllCasesAndCombosForOutput();
ret = Setup.SetCaseSelectedForOutput(ELFXName);
ret = Setup.SetCaseSelectedForOutput(ELFYName);
ret = Setup.SetCaseSelectedForOutput(RSXName);
ret = Setup.SetCaseSelectedForOutput(RSYName);
[NumberResults] = deal(integer);
[LoadCase, StepType] = deal(string_array);
[StepNum, FX, FY, FZ, MX, ParamMy, MZ, GX, GY, GZ]=deal(doublenumber);
```

Lampiran 6: RS_ScaleSF (Lanjutan)

```
[ret, NumberResults, LoadCase, ~, ~, FX, FY, ~, ~, ~, ~, ~, ~, ~] =
Results.BaseReact(NumberResults, LoadCase, StepType, StepNum, FX, FY,
FZ, MX, ParamMy, MZ, GX, GY, GZ);
for j = 1:NumberResults
if(string(LoadCase(j))) == ELFXName
ELF_FX = double(FX(j));
elseif(string(LoadCase(j))) == ELFYName
ELF_FY = double(FY(j));
elseif(string(LoadCase(j))) == RSXName
RS_FX = double(FX(j));
elseif(string(LoadCase(j))) == RSYName
RS_FY = double(FY(j));
end
end
%% Gets RS from current Response Spectrum and scales it up
[NumberLoads, Ang] = deal(integer); [LoadName, Func, CSys] =
deal(string_array); SF = doublenumber;
[ret,NumberLoads,LoadName,Func,SF,CSys,Ang] =
ResponseSpectrum.GetLoads(RSXName,NumberLoads,LoadName,Func,SF,CSys,An
g);
ScaleX = abs(ELF_FX/RS_FX)*ScaleFactor;
disp("ScaleX = " + ScaleX)
[ret, ~, ~, ~, ~, ~] =
ResponseSpectrum.SetLoads(RSXName,NumberLoads,LoadName,Func,abs(double
(SF)*ScaleX),CSys,Ang);
[ret,NumberLoads,LoadName,Func,SF,CSys,Ang] =
ResponseSpectrum.GetLoads(RSYName,NumberLoads,LoadName,Func,SF,CSys,An
g);
ScaleY = abs(ELF_FY/RS_FY)*ScaleFactor;
disp("ScaleY = " + ScaleY)
[ret, ~, ~, ~, ~, ~] =
ResponseSpectrum.SetLoads(RSYName,NumberLoads,LoadName,Func,abs(double
(SF)*ScaleY),CSys,Ang);
%% Resets eccentricity to 0.05
ret = G.API.ResponseSpectrum.SetEccentricity(RSXName,0.05);
ret = G.API.ResponseSpectrum.SetEccentricity(RSYName,0.05);
%%% replace with running the RS load cases back
ret = Analyze.RunAnalysis();
```

Lampiran 6: RS_ScaleSF (Lanjutan)

```
%%%
if ScaleY < 1 || ScaleX < 1
fprintf('Warning! Scale is smaller than 1! \n');
end
toc
end
```

Lampiran 7: RS_ECCScale

```
function RS_EccScale(G)
TableKey = "Story Max Over Avg Drifts";
DriftTable = getTableDisplay(G.API.DatabaseTables,TableKey);
condition1 = DriftTable.("Case Type")=="LinRespSpec";
condition2 = contains(DriftTable.("Output Case"),"X");
condition3 = contains(DriftTable.("Output Case"),"Y");
condition4 = contains(DriftTable.("Direction"),"X");
condition5 = contains(DriftTable.("Direction"),"Y");
RSXDriftTable = DriftTable((condition1 & condition2 & condition4),:);
RSYDriftTable = DriftTable((condition1 & condition3 & condition5),:);
RSDriftTable = [RSXDriftTable;RSYDriftTable];
TorsionalIrregularity = any(RSDriftTable.Ratio>1.2);
ExtremeTorsionalIrregularity = any(RSDriftTable.Ratio>1.4);
RSXName = unique(string(RSXDriftTable.("Output Case")));
RSYName = unique(string(RSYDriftTable.("Output Case")));
if TorsionalIrregularity || ExtremeTorsionalIrregularity
TableKey = "Story Max Over Avg Displacements";
DisplacementTable = getTableDisplay(G.API.DatabaseTables,TableKey);
condition1 = DisplacementTable.("Case Type") == "LinRespSpec";
condition2 = contains(DisplacementTable.("Output Case"),"X");
condition3 = contains(DisplacementTable.("Output Case"),"Y");
condition4 = contains(DisplacementTable.("Direction"),"X");
condition5 = contains(DisplacementTable.("Direction"),"Y");
RSXDisplacementTable = DisplacementTable((condition1 & condition2 &
condition4),:);
RSYDisplacementTable = DisplacementTable((condition1 & condition3 &
condition5),:);
AxX = (RSXDisplacementTable.Ratio./1.2).^2;
AxY = (RSYDisplacementTable.Ratio./1.2).^2;
else
AxX = 1;
AxY = 1;
end
%% Simplification by using the largest Ax for all stories
AxX = max(AxX);
AxY = max(AxY);
AxX = bound(AxX,3,1);
AxY = bound(AxY,3,1);
```

Lampiran 7: RS_ECCScale (Lanjutan)

```
ret = G.API.ResponseSpectrum.SetEccentricity(RSXName,AxX*0.05);
ret = G.API.ResponseSpectrum.SetEccentricity(RSYName,AxY*0.05);
ret = G.API.Analyze.RunAnalysis();
end
function a=bound(a,ub,lb)
a(a>ub)=ub(a>ub); a(a<lb)=lb(a<lb);
end
```

Lampiran 8: beamRebarDesign1a

```
function BeamRebarTable = beamRebarDesign1a(G)
% This function calculates the number of reinforcement needed for each
beam
% and outputs a conservative estimate of the cover and reinforcement
of
% beam ends that can be input into ETABS. This program assumes that
% concrete frame design has been done already.
% Concrete strength should be noted down in material name.
% Access DatabaseTables and DesignConcrete interfaces
DatabaseTables = G.API.DatabaseTables;
% Initializations of constants
BeamLongitudinalBars = G.Constants.BeamLongitudinalBars;
BeamTransversalBars = G.Constants.BeamTransversalBars;
ConcreteCover = G.Constants.ConcreteCover;
MinimumSpacingParallel = G.Constants.MinimumSpacingParallel;
MinimumSpacingVertical = 0.025;
MaximumLayers = G.Constants.MaximumLayers; % number of layers max
before error is given
dcOriginal = 0.06; % 60 mm from ETABS
fy = G.Constants.fy;
%% Gets beam design summary table and concrete beam properties
TableKey = "Concrete Beam Design Summary - " + G.Constants.DesignCode;
BeamDesign = getTableDisplay(DatabaseTables,TableKey,"02_SMBEAMS");
TableKey = "Frame Section Property Definitions - Concrete
Rectangular";
FraSec = getTableDisplay(DatabaseTables, TableKey);
BeamSec = FraSec(startsWith(FraSec.('Name'),"G"),:);
%%%% Classify station as ends (flexure)
for i = 1:height(BeamDesign)
MaxSta = max(BeamDesign.('Station')(BeamDesign.('UniqueName') ==
BeamDesign.('UniqueName')(i)));
MinSta = min(BeamDesign.('Station')(BeamDesign.('UniqueName') ==
BeamDesign.('UniqueName')(i)));
LnBeam = MaxSta - MinSta;
if BeamDesign.Station(i)-MinSta < 1/4*LnBeam
BeamDesign.Location(i) = cellstr('End-I');
elseif BeamDesign.Station(i)-MinSta > 3/4*LnBeam
BeamDesign.Location(i) = cellstr('End-J');
```

Lampiran 8: beamRebarDesign1a (Lanjutan)

else

```
BeamDesign.Location(i) = cellstr('Mid');
end
end
for j = 1:height(BeamSec)
%Very anti-error, results may vary
%try
BeamSec.("As Top End-
I")(j)=max(BeamDesign.("AsTopTotal")((string(BeamDesign.('DesignSect')
) == string(BeamSec.('Name')(j))) & (BeamDesign.('Location') == "End-
I")));
try
BeamSec.("As Top
Mid")(j)=max(BeamDesign.("AsTopTotal")((string(BeamDesign.('DesignSect
')) == string(BeamSec.('Name')(j))) & (BeamDesign.('Location') ==
"Mid")));
catch
BeamSec.("As Top Mid")(j) = BeamSec.("As Top End-I")(j);
end
try
BeamSec.("As Top End-J")(j) =
max(BeamDesign.("AsTopTotal")((string(BeamDesign.('DesignSect')) ==
string(BeamSec.('Name')(j))) & (BeamDesign.('Location') == "End-J")));
catch
BeamSec.("As Top End-J")(j) = BeamSec.("As Top End-I")(j);
end
BeamSec.("As Bot End-I")(j) =
max(BeamDesign.("AsBotTotal")((string(BeamDesign.('DesignSect')) ==
string(BeamSec.('Name')(j))) & (BeamDesign.('Location') == "End-I")));
try
BeamSec.("As Bot Mid")(j) =
max(BeamDesign.("AsBotTotal")((string(BeamDesign.('DesignSect')) ==
string(BeamSec.('Name')(j))) & (BeamDesign.('Location') == "Mid")));
Catch
```

Lampiran 8: beamRebarDesign1a (Lanjutan)

```
BeamSec.("As Bot Mid")(j) = BeamSec.("As Bot End-I")(j);
end
try
BeamSec.("As Bot End-J")(j) =
max(BeamDesign.("AsBotTotal")((string(BeamDesign.('DesignSect')) ==
string(BeamSec.('Name')(j))) & (BeamDesign.('Location') == "End-J")));
catch
BeamSec.("As Bot End-J")(j) = BeamSec.("As Bot End-I")(j);
end
end
% Clean BeamSec Table
BeamSec = BeamSec(:,["Name","Material","Width","Depth","As Top End-
I","As Top Mid","As Top End-J","As Bot End-I","As Bot Mid","As Bot
End-J"]);
AsTable = BeamSec(:,["As Top End-I","As Top Mid","As Top End-J","As
Bot End-I","As Bot Mid","As Bot End-J"]);
%% Table initialization
nLongBarOptions = length(BeamLongitudinalBars);
nTrvBarOptions = length(BeamTransversalBars);
% Empty final tables
AsTableCombined =
zeros(nTrvBarOptions,nLongBarOptions,height(BeamSec));
nReinTableCombined =
zeros(nTrvBarOptions,nLongBarOptions,height(BeamSec),6);
RealEffDepthCombined =
zeros(nTrvBarOptions,nLongBarOptions,height(BeamSec),6);
% Original rho and fc'
dOriginal = BeamSec.Depth - dcOriginal;
rho1 = AsTable./(dOriginal.*BeamSec.Width);
fc = str2double(extract(BeamSec.Material,digitsPattern));
FlexuralResistance1 = rho1.*fy.*(1-0.588.*rho1.*fy./fc);
for iBeamLong = 1:nLongBarOptions
for iBeamTrv = 1:nTrvBarOptions
db = BeamLongitudinalBars(iBeamLong)/1000; %convert to meters
dt = BeamTransversalBars(iBeamTrv)/1000; %convert to meters
As_single = db.^2*pi/4;
```

Lampiran 8: beamRebarDesign1a (Lanjutan)

```
nRein1 = table2array(ceil(AsTable./As_single)); nRein1 = real(nRein1);
% imaginary number handling
nRein1 = max(nRein1,2);
nRein1(nRein1(:,4) < nRein1(:,1)./2,4) = ceil(nRein1(nRein1(:,4) <
nRein1(:,1)./2,1)./2);
ReinMax = max(nRein1(:,1),nRein1(:,3));
nRein1(nRein1(:,6) < nRein1(:,3)./2,6) = ceil(nRein1(nRein1(:,6) <
nRein1(:,3)./2,3)./2);
nRein1(nRein1(:,5) < ReinMax./4,5) = ceil(nRein1(nRein1(:,5) <
ReinMax./4,2)./4);
MaxPerLayer=floor((BeamSec.("Width")-2*ConcreteCover-
2.*dt+MinimumSpacingParallel)./(db+MinimumSpacingParallel));
nLayers = ceil(nRein1./MaxPerLayer);
conservative_d = nLayers;
MaximumLayers = max(nLayers,[],[1 2]); %overwrite max layers
CoverOptions = zeros(1,MaximumLayers);
LayerDistance = zeros(1,MaximumLayers);
for iCoverOpt = 1:MaximumLayers
CoverOptions(iCoverOpt) = ConcreteCover + dt + db*iCoverOpt/2 +
MinimumSpacingVertical*(iCoverOpt-1)/2;
LayerDistance(iCoverOpt) = ConcreteCover + dt + db*(1+(2*iCoverOpt-
2))/2 + MinimumSpacingVertical*(iCoverOpt-1);
end
conservative_d = (CoverOptions(conservative_d));
conservative_d = BeamSec.Depth-conservative_d;
findzeros = sum(sum(conservative_d == 0));
if findzeros > 0
error = "Too many layers, increase max layers"
return
end
% Recalculate As with new d
FlexuralResistance2 =
table2array(FlexuralResistance1.*(dOriginal./conservative_d).^2);
rho2 = (125.*fc.*( 1 - ((-294.*FlexuralResistance2 +
125.*fc)./(125.*fc)).^(1/2)))./(147.*fy);
As2 = rho2.*BeamSec.Width.*(conservative_d);
nRein2 = ceil(As2./As_single); nRein2 = real(nRein2); % imaginary
number handling
```

Lampiran 8: beamRebarDesign1a (Lanjutan)

```
% Count total As
TotalAs = sum(nRein2,2).*As_single;
LastLayerRein = mod(nRein2,MaxPerLayer);
tempMaxPerLayer = repmat(MaxPerLayer,1,6);
LastLayerRein(LastLayerRein == 0) = tempMaxPerLayer(LastLayerRein==0);
% Real_d = %%
RealEffDepth = nLayers;
for iCover = 1:MaximumLayers
RealEffDepth(nLayers == iCover) = (mean(LayerDistance(1:max([iCover-
1,1]))).*tempMaxPerLayer(nLayers == iCover).*(iCover-1) +
…LayerDistance(iCover).*LastLayerRein(nLayers ==
iCover))./nRein2(nLayers==iCover);
end
% Finalize Results
FinalTable = BeamSec(:,["Name","Material","Width","Depth"]);
FinalTable.("Long. Bar")(:) = db*1000;
FinalTable.("Trv. Bar")(:) = dt*1000;
FinalTable.("Total As") = TotalAs;
AsTableCombined(iBeamTrv,iBeamLong,:) = TotalAs;
nReinTableCombined(iBeamTrv,iBeamLong,:,:) = nRein2;
RealEffDepthCombined(iBeamTrv,iBeamLong,:,:) = RealEffDepth;
end
end
%% Get minimum reinforcement
%(iBeamTrv,iBeamLong,As)
[~, idx1] = min(AsTableCombined,[],[1 2],"linear"); %replace ~ with
AsMinimum for more details
s[var1,var2,~] =
ind2sub([nTrvBarOptions,nLongBarOptions,height(BeamSec)],idx1);
var1 = reshape(var1,[numel(var1) 1]);
var2 = reshape(var2,[numel(var2) 1]);
idx1 = reshape(idx1,[numel(idx1) 1]);
idx1 = idx1 + numel(idx1).*nTrvBarOptions.*nLongBarOptions.*(0:5);
dbMinimum = transpose(BeamLongitudinalBars(1,var2));
dtMinimum = transpose(BeamTransversalBars(1,var1));
nReinMinimum = nReinTableCombined(idx1);
RealEffDepthMinimum = RealEffDepthCombined(idx1);
AsProv = nReinMinimum.*((dbMinimum./1000).^2).*pi./4;
```

Lampiran 8: beamRebarDesign1a (Lanjutan)

```
Name = BeamSec.Name;
BeamRebarTable =
table(Name,dbMinimum,dtMinimum,nReinMinimum,RealEffDepthMinimum,AsProv
);
```

Lampiran 9: setRebar1_1

```
function ret = setRebar1_1(G,BeamRebarTable)
%%
ret = G.API.SapModel.SetModelIsLocked(false);
Name = "";
MatPropLong = "";
MatPropConfine = "";
CoverTop = -1;
CoverBot = -1;
TopLeftArea = -1;
TopRightArea = -1;
BotLeftArea = -1;
BotRightArea = -1;
CoverInput = BeamRebarTable.RealEffDepthMinimum;
TopCoverInput = max(CoverInput(:,1:3),[],2);
BottomCoverInput = max(CoverInput(:,4:6),[],2);
AsReinforcement = BeamRebarTable.AsProv;
for i = 1:height(BeamRebarTable)
Name = string(BeamRebarTable.Name(i));
[ret, MatPropLong, MatPropConfine, CoverTop, CoverBot, …
TopLeftArea, TopRightArea, BotLeftArea, BotRightArea] =
G.API.PropFrame.GetRebarBeam( ...
Name, MatPropLong, MatPropConfine, CoverTop, CoverBot, ...
TopLeftArea, TopRightArea, BotLeftArea, BotRightArea);
TopLeftArea = AsReinforcement(i,1);
TopRightArea = AsReinforcement(i,3);
BotLeftArea = AsReinforcement(i,4);
BotRightArea = AsReinforcement(i,6);
CoverTop = TopCoverInput(i);
CoverBot = BottomCoverInput(i);
ret = G.API.PropFrame.SetRebarBeam( …
Name, MatPropLong, MatPropConfine, CoverTop, CoverBot, ...
TopLeftArea, TopRightArea, BotLeftArea, BotRightArea);
end
```

Lampiran 10: columnRebarDesign1a

```
function TopColumnReinTable = columnRebarDesign1a(G)
% This function gets the As required from "To Be Designed" columns.
% TopColumnReinTable = columnRebarDesign1a(G)
% Applies to top columns only (04_TOPCOLUMNS)
% Access DatabaseTables and DesignConcrete interfaces
DatabaseTables = G.API.DatabaseTables;
ColumnLongitudinalBars = G.Constants.ColumnLongitudinalBars;
ColumnLongitudinalBarsArea = (ColumnLongitudinalBars./1000).^2*pi./4;
% Gets beam design summary table
TableKey = "Concrete Column Design Summary - " +
G.Constants.DesignCode;
ColumnDesign=getTableDisplay(DatabaseTables,TableKey,"04_TOPCOLUMNS");
TableKey = "Frame Section Property Definitions - Concrete
Rectangular";
FraSec = getTableDisplay(DatabaseTables, TableKey);
Name = unique(ColumnDesign.DesignSect);
ColSec = innerjoin(FraSec,table(Name));
%% Column max As required by design
AsReq = zeros(height(ColSec),1);
for i=1:height(ColSec)
AsReq(i)= max(ColumnDesign.("As")((string(ColumnDesign.('DesignSect'))
== string(ColSec.('Name')(i)))));
end
ColSec.("As req.") = AsReq;
TopColumnReinTable = ColSec(:,["Name","As req.","Depth","Width"]);
MaxAsVector =
0.06.*TopColumnReinTable.Depth.*TopColumnReinTable.Width; % Overwrite
large As to 6%
TopColumnReinTable.("As req.")(TopColumnReinTable.("As
req.")>MaxAsVector) = MaxAsVector(TopColumnReinTable.("As
req.")>MaxAsVector);
```

Lampiran 11: columnRebarDesign1b

```
function ColumnRebarTable = columnRebarDesign1b(G, TopColumnReinTable)
% This function gets the As required from "To Be Designed" columns.
% TopColumnReinTable = columnRebarDesign1b(G)
% Applies to all columns
% Access DatabaseTables and DesignConcrete interfaces
DatabaseTables = G.API.DatabaseTables;
DesignConcrete = G.API.DesignConcrete;
% Gets beam design summary table
TableKey = "Concrete Column Design Summary - " +
G.Constants.DesignCode;
ColumnDesign = getTableDisplay(DatabaseTables,TableKey,"01_SMF");
TableKey = "Frame Section Property Definitions - Concrete
Rectangular";
FraSec = getTableDisplay(DatabaseTables, TableKey);
Name = unique(ColumnDesign.DesignSect);
ColSec = innerjoin(FraSec,table(Name));
%% Column max As required by design
AsReq = zeros(height(ColSec),1);
for i=1:height(ColSec)
AsReq(i)=max(ColumnDesign.("As")((string(ColumnDesign.('DesignSect'))
== string(ColSec.('Name')(i)))));
end
ColSec.("As req.") = AsReq;
ColumnRebarTable = ColSec(:,["Name","As req.","Depth","Width"]);
MaxAsVector = 0.06.*ColumnRebarTable.Depth.*ColumnRebarTable.Width; %
Overwrite large As to 6%
ColumnRebarTable.("As req.")(ColumnRebarTable.("As req.")>MaxAsVector)
= MaxAsVector(ColumnRebarTable.("As req.")>MaxAsVector);
[~,idx2]=ismember(TopColumnReinTable.Name,ColumnRebarTable.Name);
ColumnRebarTable(idx2,:) = TopColumnReinTable;
```

Lampiran 12: setRebar2_1

```
function ret = setRebar2_1(G,ColumnRebarTable)
% This function applies column reinforcing to the model without using
% DatabaseTables.
%% Column
MatPropLong = "";
MatPropConfine = "";
Pattern = -1;
ConfineType = -1;
Cover = -1;
NumberCBars = -1;
NumberR3Bars = -1;
NumberR2Bars = -1;
RebarSize = "";
TieSize = "";
TieSpacingLongit = -1;
Number2DirTieBars = -1;
Number3DirTieBars = -1;
ToBeDesigned = false;
LongitCornerRebarSize = "";
LongitRebarArea = -1;
LongitCornerRebarArea = -1;
ret = G.API.SapModel.SetModelIsLocked(false);
%% Major shear - V2
% Minor shear - V3
% h (Depth) is parallel to 2 axis
% b (Width) is parallel to 3 axis
% Number of rebar is opposite of b & h
% (Maj --> Number of Bars 3-dir and vice-versa)
for i = 1:height(ColumnRebarTable)
Name = string(ColumnRebarTable.Name(i));
[ret, MatPropLong, MatPropConfine, Pattern, ...
ConfineType, Cover, NumberCBars, ...
NumberR3Bars, NumberR2Bars, RebarSize, ...
TieSize, TieSpacingLongit, Number2DirTieBars,...
Number3DirTieBars, ToBeDesigned, ...
LongitCornerRebarSize, LongitRebarArea, ...
LongitCornerRebarArea] = G.API.PropFrame.GetRebarColumn_1(...
Name, MatPropLong, MatPropConfine, Pattern, ...
```

Lampiran 12: setRebar2_1 (Lanjutan)

```
ConfineType, Cover, NumberCBars, ...
NumberR3Bars, NumberR2Bars, RebarSize, ...
TieSize, TieSpacingLongit, Number2DirTieBars,...
Number3DirTieBars, ToBeDesigned, ...
LongitCornerRebarSize, LongitRebarArea, ...
LongitCornerRebarArea);
% New rebar
TotalRebar = ceil(ColumnRebarTable.("As req.")(i)./(LongitRebarArea));
dir2rein_calc=ceil((2 +
TotalRebar./2)./(1+ColumnRebarTable.Width(i)./ColumnRebarTable.Depth(i
)));
dir3rein_calc=(ColumnRebarTable.Width(i)./ColumnRebarTable.Depth(i)).*
dir2rein_calc;
NumberR2Bars = ceil(dir2rein_calc);
NumberR3Bars = ceil(dir3rein_calc);
ToBeDesigned = false;
ret = G.API.PropFrame.SetRebarColumn(...
Name, MatPropLong, MatPropConfine, Pattern, ...
ConfineType, Cover, NumberCBars, ...
NumberR3Bars, NumberR2Bars, RebarSize, ...
TieSize, TieSpacingLongit, Number2DirTieBars,...
Number3DirTieBars, ToBeDesigned);
end
```

Lampiran 13: columnAsDesign

```
function [ColumnReinforcementTable, RedesignFlag] =
columnAsDesign(G,AssignFlag)
if nargin < 2
AssignFlag = true;
end
RedesignFlag = true; %generally true
TableKey = "Concrete Column Design Summary - " +
G.Constants.DesignCode;
ColumnResultsTable =
getTableDisplay(G.API.DatabaseTables,TableKey,"01_SMF");
TableKey = "Concrete Joint Design Summary - " +
G.Constants.DesignCode;
JointResultsTable =
getTableDisplay(G.API.DatabaseTables,TableKey,"01_SMF");
TopJointResultsTable =
getTableDisplay(G.API.DatabaseTables,TableKey,"04_TOPCOLUMNS");
BCCJointResultsTable =
JointResultsTable(~ismember(JointResultsTable,TopJointResultsTable),:)
; %remove top columns
% Turn off redesign if JS
if sum([JointResultsTable.JSMajRatio > 1; JointResultsTable.JSMinRatio
> 1])
RedesignFlag = false;
end
TableKey = "Frame Section Property Definitions - Concrete
Rectangular";
ConcreteSectionsTable=getTableDisplay(G.API.DatabaseTables,TableKey,"0
1_SMF");
TableKey = "Frame Section Property Definitions - Concrete Column
Reinforcing";
ColumnReinforcementTable =
getTableDisplay(G.API.DatabaseTables,TableKey,"01_SMF");
%% For column PMM and rho
ColumnTable=ColumnReinforcementTable(startsWith(ColumnReinforcementTab
le.Name, "K"),"Name");
ColumnReinforcementTable =
innerjoin(ColumnTable,ColumnReinforcementTable);
```

Lampiran 13: columnAsDesign (Lanjutan)

```
ColumnReinforcementTable = ColumnReinforcementTable(:,["Name","Number
Bars 2-Dir","Number Bars 3-Dir","Longitudinal Bar Size"]);
ColumnNumberOfLongitudinalBars = (ColumnReinforcementTable.("Number
Bars 3-Dir")+ColumnReinforcementTable.("Number Bars 2-Dir")-2).*2;
ColumnAs=ColumnNumberOfLongitudinalBars.*(ColumnReinforcementTable.("L
ongitudinal Bar Size")./1000).^2*pi./4;
ColumnSectionsTable = innerjoin(ColumnTable,ConcreteSectionsTable);
ColumnSectionsTable = ColumnSectionsTable(:,["Name","Depth","Width"]);
ColumnAg = ColumnSectionsTable.Depth.*ColumnSectionsTable.Width;
ColumnRho = ColumnAs./ColumnAg.*100;
ColumnPMM = zeros(height(ColumnReinforcementTable),1);
for i=1:height(ColumnReinforcementTable)
ColumnPMM(i)=max(ColumnResultsTable.PMMRatio(ColumnReinforcementTable.
Name(i) == string(ColumnResultsTable.DesignSect)));
end
ColumnReinforcementTable.PMM = ColumnPMM;
%% For column BCC ratio and JS ratio
BCCJointResultsTable =
BCCJointResultsTable(:,["DesignSect","BCMajRatio","BCMinRatio","JSMajR
atio","JSMinRatio"]);
JointResultsTable =
JointResultsTable(:,["DesignSect","BCMajRatio","BCMinRatio","JSMajRati
o","JSMinRatio"]);
JointResultsTable.BCMajRatio(~ismember(JointResultsTable.DesignSect,BC
CJointResultsTable.DesignSect),:) = NaN; %remove top columns
JointResultsTable.BCMinRatio(~ismember(JointResultsTable.DesignSect,BC
CJointResultsTable.DesignSect),:) = NaN;
JointResultsTable = renamevars(JointResultsTable,"DesignSect","Name");
ColumnReinforcementTable =
innerjoin(ColumnReinforcementTable,JointResultsTable);
ColumnReinforcementTable =
innerjoin(ColumnReinforcementTable,ColumnSectionsTable);
%% New As (algorithm required)
NewColumnRho =
max([ColumnReinforcementTable.PMM,ColumnReinforcementTable.BCMajRatio,
ColumnReinforcementTable.BCMinRatio],[],2).^1.2.*ColumnRho; %sqrt of
PMM ratio
```

Lampiran 13: columnAsDesign (Lanjutan)

```
NewColumnRho(NewColumnRho>6) = 6; NewColumnRho(NewColumnRho<1) = 1;
%bind to 1% to 6%
NewColumnRho(NewColumnRho>0.95.*ColumnRho & NewColumnRho<ColumnRho) =
ColumnRho(NewColumnRho>0.95.*ColumnRho & NewColumnRho<ColumnRho);
%Change rho if ratio is less than 0.95
NewColumnAs = ColumnAg.*NewColumnRho./100;
ColumnReinforcementTable.Rho = ColumnRho;
ColumnReinforcementTable.NewRho = NewColumnRho;
ColumnReinforcementTable.NewAs = NewColumnAs;
%% Number of rebar required
ColumnLongitudinalBars = G.Constants.ColumnLongitudinalBars./1000;
ColumnLongitudinalBarsArea = (ColumnLongitudinalBars).^2.*pi./4;
NumberOfRebar = ceil(NewColumnAs./ColumnLongitudinalBarsArea./2)*2;
dir2rein_calc = (2 +
NumberOfRebar./2)./(1+ColumnReinforcementTable.Width./ColumnReinforcem
entTable.Depth);
dir3rein_calc =
(ColumnReinforcementTable.Width./ColumnReinforcementTable.Depth).*dir2
rein_calc;
NumberR2Bars = round(dir2rein_calc);
NumberR3Bars = round(dir3rein_calc);
NumberR2BarsMax = floor((ColumnReinforcementTable.Depth-
G.Constants.ConcreteCover.*2-
max(G.Constants.ColumnTransversalBars)./1000.*2+0.04)./(0.04+ColumnLon
gitudinalBars));
NumberR3BarsMax=floor((ColumnReinforcementTable.Width-
G.Constants.ConcreteCover.*2-
max(G.Constants.ColumnTransversalBars)./1000.*2+0.04)./(0.04+ColumnLon
gitudinalBars));
NumberR2BarsOld = NumberR2Bars;
NumberR3BarsOld = NumberR3Bars;
NumberR2Bars(NumberR2Bars>NumberR2BarsMax) = Inf;
NumberR3Bars(NumberR3Bars>NumberR3BarsMax) = Inf;
% Create logical index arrays
% If rho is constrained by spacing requirements, take the largest
diameter.
idxR2 = all(NumberR2Bars == Inf, 2);
idxR3 = all(NumberR3Bars == Inf, 2);
```

Lampiran 13: columnAsDesign (Lanjutan)

```
% Update the rows based on the logical indices
NumberR2Bars(idxR2,end) = NumberR2BarsOld(idxR2,end);
NumberR3Bars(idxR3,end) = NumberR3BarsOld(idxR3,end);
NumberR2BarsPick(:,:,1) = NumberR2Bars;
NumberR2BarsPick(:,:,2) = NumberR2BarsMax;
NumberR2BarsProv = min(NumberR2BarsPick,[],3);
NumberR3BarsPick(:,:,1) = NumberR3Bars;
NumberR3BarsPick(:,:,2) = NumberR3BarsMax;
NumberR3BarsProv = min(NumberR3BarsPick,[],3);
NumberRebarProv = (NumberR2BarsProv + NumberR3BarsProv - 2).*2;
AsProv = NumberRebarProv.*ColumnLongitudinalBarsArea;
[~,idx] = min(AsProv,[],2);
LongitudinalRebarPick =
reshape(G.Constants.ColumnLongitudinalBars(idx),[height(idx),1]);
NumberRebarPick = reshape(arrayfun(@(i) NumberRebarProv(i, idx(i)),
1:height(idx)),[height(idx),1]);
NewNumberR2BarsPick = reshape(arrayfun(@(i) NumberR2BarsProv(i,
idx(i)), 1:height(idx)),[height(idx),1]);
NewNumberR3BarsPick = reshape(arrayfun(@(i) NumberR3BarsProv(i,
idx(i)), 1:height(idx)),[height(idx),1]);
ColumnReinforcementTable.RebarSize = LongitudinalRebarPick;
ColumnReinforcementTable.NewNumberOfBars = NumberRebarPick;
ColumnReinforcementTable.("New Number Bars 2-Dir") =
NewNumberR2BarsPick;
ColumnReinforcementTable.("New Number Bars 3-Dir") =
NewNumberR3BarsPick;
if AssignFlag
ret = setRebar3_1(G,ColumnReinforcementTable);
end
```

Lampiran 14: columnAvDesign

```
function ColumnTable = columnAvDesign(G)
TableKey = "Concrete Column Design Summary - " +
G.Constants.DesignCode;
ColumnResultsTable=getTableDisplay(G.API.DatabaseTables,TableKey,"01_S
MF");
TableKey = "Frame Section Property Definitions - Concrete
Rectangular";
ConcreteSectionsTable =
getTableDisplay(G.API.DatabaseTables,TableKey,"01_SMF");
TableKey = "Frame Section Property Definitions - Concrete Column
Reinforcing";
ColumnReinforcementTable =
getTableDisplay(G.API.DatabaseTables,TableKey,"01_SMF");
TableKey = "Frame Assignments - Summary";
FrameAssignmentsTable = getTableDisplay(G.API.DatabaseTables,
TableKey, "01_SMF");
TableKey = "Material Properties - Concrete Data";
ConcreteDataTable = getTableDisplay(G.API.DatabaseTables, TableKey);
ConcreteDataTable = ConcreteDataTable(:,["Material", "Fc"]);
% Table for Pu
ret=G.API.DatabaseTables.SetLoadCasesSelectedForDisplay(NET.createArra
y('System.String',2));
ret=G.API.DatabaseTables.SetLoadCombinationsSelectedForDisplay(NET.cre
ateArray('System.String',2));
TableKey = "Element Forces - Columns";
LoadCombo = NET.createArray('System.String',2);
LoadCombo(1) = "COMBMAX";
ret=G.API.DatabaseTables.SetLoadCombinationsSelectedForDisplay(LoadCom
bo);
ColumnForcesTable = getTableDisplay(G.API.DatabaseTables, TableKey,
"01_SMF");
% Assign analysis section to ColumnForcesTable
ColumnAssignments = FrameAssignmentsTable(:,["UniqueName","Analysis
Section","Length"]);
ColumnAssignments = renamevars(ColumnAssignments,"UniqueName","Unique
Name");
ColumnForcesTable = innerjoin(ColumnForcesTable,ColumnAssignments);
```

Lampiran 14: columnAvDesign (Lanjutan)

```
ColumnAssignments = renamevars(ColumnAssignments,"Analysis
Section","Name");
% Determining rho
ColumnTable=ColumnReinforcementTable(startsWith(ColumnReinforcementTab
le.Name, "K"),"Name");
ColumnReinforcementTable=innerjoin(ColumnTable,ColumnReinforcementTabl
e);
ColumnReinforcementTable = ColumnReinforcementTable(:,["Name","Number
Bars 2-Dir","Number Bars 3-Dir","Longitudinal Bar Size"]);
ColumnNumberOfLongitudinalBars = (ColumnReinforcementTable.("Number
Bars 3-Dir")+ColumnReinforcementTable.("Number Bars 2-Dir")-2).*2;
ColumnAs=ColumnNumberOfLongitudinalBars.*(ColumnReinforcementTable.("L
ongitudinal Bar Size")./1000).^2*pi./4;
ColumnSectionsTable = innerjoin(ColumnTable,ConcreteSectionsTable);
ColumnSectionsTable =
ColumnSectionsTable(:,["Name","Material","Depth","Width"]);
ColumnAg = ColumnSectionsTable.Depth.*ColumnSectionsTable.Width;
ColumnRho = ColumnAs./ColumnAg.*100;
% Join tables
ColumnReinforcementTable =
innerjoin(ColumnReinforcementTable,ColumnSectionsTable);
ColumnReinforcementTable =
innerjoin(ColumnReinforcementTable,ConcreteDataTable);
% Determining large axial flag
PuVector = zeros(height(ColumnTable),1);
VMajRebarVector = zeros(height(ColumnTable),1);
VMinRebarVector = zeros(height(ColumnTable),1);
for i = 1:height(ColumnTable)
VMajRebarVector(i,1)=max(ColumnResultsTable.VMajRebar(ColumnResultsTab
le.DesignSect == string(ColumnTable.("Name")(i))));
VMinRebarVector(i,1)=max(ColumnResultsTable.VMinRebar(ColumnResultsTab
le.DesignSect == string(ColumnTable.("Name")(i))));
PuVector(i) = min(ColumnForcesTable.P(ColumnForcesTable.("Analysis
Section")==string(ColumnTable.("Name")(i)))); %% in kN
end
LargeAxialFlagVector = 0.3.*ColumnAg.*G.Constants.fcColumn.*10^3 <
abs(PuVector);
%% Shear rebar minimization
```

Lampiran 14: columnAvDesign (Lanjutan)

```
ColumnLongitudinalBars = G.Constants.ColumnLongitudinalBars./1000;
ColumnLongitudinalBarsArea = (ColumnLongitudinalBars).^2.*pi./4;
NumberOfRebar = ceil(ColumnAs./ColumnLongitudinalBarsArea./2)*2;
ColumnTransversalBars = G.Constants.ColumnTransversalBars./1000;
ColumnTransversalBarsArea = (ColumnTransversalBars).^2.*pi./4;
dir2rein_calc =
(2+NumberOfRebar./2)./(1+ColumnReinforcementTable.Width./ColumnReinfor
cementTable.Depth);
dir3rein_calc =
(ColumnReinforcementTable.Width./ColumnReinforcementTable.Depth).*dir2
rein_calc;
NumberR2Bars = round(dir2rein_calc);
NumberR3Bars = round(dir3rein_calc);
NumberR2BarsMax = floor((ColumnReinforcementTable.Depth-
G.Constants.ConcreteCover.*2-
max(G.Constants.ColumnTransversalBars)./1000.*2+0.04)./(0.04+ColumnLon
gitudinalBars));
NumberR3BarsMax = floor((ColumnReinforcementTable.Width-
G.Constants.ConcreteCover.*2-
max(G.Constants.ColumnTransversalBars)./1000.*2+0.04)./(0.04+ColumnLon
gitudinalBars));
NumberR2BarsOld = NumberR2Bars;
NumberR3BarsOld = NumberR3Bars;
NumberR2Bars(NumberR2Bars>NumberR2BarsMax) = Inf;
NumberR3Bars(NumberR3Bars>NumberR3BarsMax) = Inf;
% Create logical index arrays
% If rho is constrained by spacing requirements, take the largest
diameter.
idxR2 = all(NumberR2Bars == Inf, 2);
idxR3 = all(NumberR3Bars == Inf, 2);
% Update the rows based on the logical indices
NumberR2Bars(idxR2,end) = NumberR2BarsOld(idxR2,end);
NumberR3Bars(idxR3,end) = NumberR3BarsOld(idxR3,end);
% Minimum legs of shear reinforcement / T2Bars means parallel to 2 /
% revised (check this)
NumberT2BarsMin = ceil((1+NumberR3Bars)./2);
NumberT2BarsMin(LargeAxialFlagVector, :) =
NumberR3Bars(LargeAxialFlagVector, :);
```

Lampiran 14: columnAvDesign (Lanjutan)

```
NumberT2BarsMax = NumberR3Bars;
NumberT3BarsMin = ceil((1+NumberR2Bars)./2);
NumberT3BarsMin(LargeAxialFlagVector, :) =
NumberR2Bars(LargeAxialFlagVector, :);
NumberT3BarsMax = NumberR2Bars;
SpacingMax=min([1/4.*min([ColumnReinforcementTable.Depth,ColumnReinfor
cementTable.Width],[],2),0.1.*ones([height(ColumnTable),1]),6.*ColumnR
einforcementTable.("Longitudinal Bar Size")],[],2);
% Ash calculations
ColumnAch = (ColumnReinforcementTable.Depth-
2*G.Constants.ConcreteCover).*(ColumnReinforcementTable.Width-
2*G.Constants.ConcreteCover);
FcVector = ColumnReinforcementTable.Fc;
kfVector =
max([FcVector./1000./175+0.6,ones(height(ColumnTable),1)],[],2);
% nLongRebarVector changeable: assume minimum for maximum kN
nLongRebarVector = (NumberT2BarsMin+NumberT3BarsMin-2).*2;
knVector = nLongRebarVector./(nLongRebarVector-2);
Ash1Vector = 0.3.*(ColumnAg./ColumnAch -
1).*FcVector./1000./G.Constants.fyt;
Ash2Vector = 0.09.*(FcVector./1000)./G.Constants.fyt;
Ash3Vector =
0.2.*kfVector.*knVector.*abs(PuVector)./(G.Constants.fyt.*ColumnAch)./
10^3; % to unitless
Ash1Vector = repmat(Ash1Vector,1,length(ColumnLongitudinalBars));
Ash2Vector = repmat(Ash2Vector,1,length(ColumnLongitudinalBars));
AshVector = max(cat(3,Ash1Vector,Ash2Vector,Ash3Vector),[],3); % Each
longitudinal bar gets its own column
VMajRebarConfinementVector=AshVector.*ColumnReinforcementTable.Depth;
VMinRebarConfinementVector=AshVector.*ColumnReinforcementTable.Width;
VMajRebarVector=repmat(VMajRebarVector,1,length(ColumnLongitudinalBars
));
VMinRebarVector=repmat(VMinRebarVector,1,length(ColumnLongitudinalBars
));
VMajRebarFinalVector =
max(cat(3,VMajRebarVector,VMajRebarConfinementVector),[],3);
VMinRebarFinalVector =
max(cat(3,VMinRebarVector,VMinRebarConfinementVector),[],3);
```

Lampiran 14: columnAvDesign (Lanjutan)

```
SpacingOptions =
(0.05:G.Constants.MultipleSpacingTrv:max(SpacingMax,[],[1 2]));
%Minimum 50 mm, maximum 150 mm
TotalSteel =
zeros(height(ColumnReinforcementTable),length(ColumnLongitudinalBars),
length(ColumnTransversalBars),length(SpacingOptions));
% Take the same empty matrix
NumberOfLegsMajMatrix = TotalSteel;
NumberOfLegsMinMatrix = TotalSteel;
NumberR2BarsMatrix = TotalSteel;
NumberR3BarsMatrix = TotalSteel;
SpacingProv = TotalSteel;
for i = 1:length(ColumnLongitudinalBars)
for j = 1:length(ColumnTransversalBars)
for k = 1:length(SpacingOptions)
NumberR3BarsTemp = NumberR3Bars(:,i);
NumberR2BarsTemp = NumberR2Bars(:,i);
As = NumberOfRebar(:,i).*ColumnLongitudinalBarsArea(1,i);
NumberOfLegsMaj=ceil(VMajRebarFinalVector(:,i)./ColumnTransversalBarsA
rea(1,j).*SpacingOptions(k));
NumberOfLegsMin=ceil(VMinRebarFinalVector(:,i)./ColumnTransversalBarsA
rea(1,j).*SpacingOptions(k));
NumberOfLegsMajOld = NumberOfLegsMaj;
NumberOfLegsMinOld = NumberOfLegsMin;
NumberOfLegsMaj(NumberOfLegsMaj>NumberR3BarsMax(:,i)) = Inf; %sets to
infinity
NumberOfLegsMaj(NumberOfLegsMaj<NumberT2BarsMin(:,i)) =
NumberT2BarsMin(NumberOfLegsMaj<NumberT2BarsMin(:,i),i);
NumberR3BarsTemp(NumberOfLegsMaj>NumberR3Bars(:,i)) =
NumberOfLegsMaj(NumberOfLegsMaj>NumberR3Bars(:,i));
% New minimum legs
NumberT2BarsMax = NumberR2Bars;
NumberOfLegsMin(NumberOfLegsMin>NumberR2BarsMax(:,i)) = Inf;
NumberOfLegsMin(NumberOfLegsMin<NumberT3BarsMin(:,i)) =
NumberT3BarsMin(NumberOfLegsMin<NumberT3BarsMin(:,i),i);
```

Lampiran 14: columnAvDesign (Lanjutan)

```
NumberR2BarsTemp(NumberOfLegsMin>NumberR2Bars(:,i)) =
NumberOfLegsMin(NumberOfLegsMin>NumberR2Bars(:,i));
NumberOfLegsMaj(SpacingMax<SpacingOptions(k)) = Inf;
NumberOfLegsMin(SpacingMax<SpacingOptions(k)) = Inf;
% Length T3 means parallel to 3
LengthT2 =
NumberOfLegsMaj.*ColumnTransversalBarsArea(j).*(ColumnReinforcementTab
le.Depth-(2.*G.Constants.ConcreteCover-ColumnTransversalBars(j)));
LengthT3=NumberOfLegsMin.*ColumnTransversalBarsArea(j).*(ColumnReinfor
cementTable.Width-(2.*G.Constants.ConcreteCover-
ColumnTransversalBars(j)));
NumberOfLegsMajMatrix(:,i,j,k) = NumberOfLegsMaj;
NumberOfLegsMinMatrix(:,i,j,k) = NumberOfLegsMin;
NumberR2BarsMatrix(:,i,j,k) = NumberR2BarsTemp;
NumberR3BarsMatrix(:,i,j,k) = NumberR3BarsTemp;
NewAs=(NumberR2BarsTemp+NumberR3BarsTemp-
2).*2.*ColumnLongitudinalBarsArea(1,i);
TotalSteel(:,i,j,k) = NewAs+(LengthT2+LengthT3)./SpacingOptions(k);
end
end
end
[~,idx] = min(TotalSteel,[],[2 3 4],'linear');
NumberOfLegsMajUse = NumberOfLegsMajMatrix(idx);
NumberOfLegsMinUse = NumberOfLegsMinMatrix(idx);
NumberR2BarsUse = NumberR2BarsMatrix(idx);
NumberR3BarsUse = NumberR3BarsMatrix(idx);
[idx1,idx2,idx3,idx4] = ind2sub(size(TotalSteel),idx);
DiameterTransversalUse = zeros(height(idx),1);
DiameterLongitudinalUse = DiameterTransversalUse;
SpacingUse = DiameterTransversalUse;
NumberR2BarsUseOld = DiameterTransversalUse;
NumberR3BarsUseOld = DiameterTransversalUse;
for i = 1:length(idx1)
    DiameterLongitudinalUse(i) = ColumnLongitudinalBars(idx2(i));
    DiameterTransversalUse(i) = ColumnTransversalBars(idx3(i));
    SpacingUse(i) = SpacingOptions(idx4(i));
    NumberR2BarsUseOld(i) = NumberR2Bars(i,idx2(i));
    NumberR3BarsUseOld(i) = NumberR3Bars(i,idx2(i));
```

Lampiran 14: columnAvDesign (Lanjutan)

```
end
    ColumnTable.("RebarSize") = DiameterLongitudinalUse.*1000;
    ColumnTable.("New Number Bars 2-Dir") = NumberR2BarsUse;
    ColumnTable.("Old Number Bars 2-Dir") = NumberR2BarsUseOld;
    ColumnTable.("New Number Bars 3-Dir") = NumberR3BarsUse;
    ColumnTable.("Old Number Bars 3-Dir") = NumberR3BarsUseOld;
    ColumnTable.DiameterTransversal = DiameterTransversalUse.*1000;
    ColumnTable.NumberOfLegsMaj = NumberOfLegsMajUse;
    ColumnTable.NumberOfLegsMin = NumberOfLegsMinUse;
    ColumnTable.SpacingUse = SpacingUse;
    ColumnTable.As = (NumberR2BarsUse + NumberR3BarsUse -
2).*(DiameterLongitudinalUse.^2*pi/4);
    ColumnTable = innerjoin(ColumnTable,ColumnAssignments);
    ColumnTable = innerjoin(ColumnTable,ColumnSectionsTable);
```

Lampiran 15: setRebar3_1

```
function [ret] = setRebar3_1(G,ColumnReinforcementTable)
%% Works in conjunction with columnAsDesign
ret = G.API.SapModel.SetModelIsLocked(false);
% Initialization
MatPropLong = "";
MatPropConfine = "";
Pattern = -1;
ConfineType = -1;
Cover = -1;
NumberCBars = -1;
NumberR3Bars = -1;
NumberR2Bars = -1;
RebarSize = "";
TieSize = "";
TieSpacingLongit = -1;
Number2DirTieBars = -1;
Number3DirTieBars = -1;
ToBeDesigned = false;
LongitCornerRebarSize = "";
LongitRebarArea = -1;
LongitCornerRebarArea = -1;
%% Design
for i = 1:height(ColumnReinforcementTable)
Name = string(ColumnReinforcementTable.Name(i));
% Get rebar
[ret1, MatPropLong, MatPropConfine, Pattern, ...
ConfineType, Cover, NumberCBars, ...
NumberR3Bars, NumberR2Bars, RebarSize, ...
TieSize, TieSpacingLongit, Number2DirTieBars,...
Number3DirTieBars, ToBeDesigned, ...
LongitCornerRebarSize, LongitRebarArea, ...
LongitCornerRebarArea] = G.API.PropFrame.GetRebarColumn_1(...
Name, MatPropLong, MatPropConfine, Pattern, ...
ConfineType, Cover, NumberCBars, ...
NumberR3Bars, NumberR2Bars, RebarSize, ...
TieSize, TieSpacingLongit, Number2DirTieBars,...
Number3DirTieBars, ToBeDesigned, ...
LongitCornerRebarSize, LongitRebarArea, ...
```

Lampiran 15: setRebar3_1 (Lanjutan)

```
LongitCornerRebarArea);
% New rebar (BCC design)
% Redistribute
NumberR2Bars = ColumnReinforcementTable.("New Number Bars 2-Dir")(i);
NumberR3Bars = ColumnReinforcementTable.("New Number Bars 3-Dir")(i);
RebarSize = string(ColumnReinforcementTable.RebarSize(i));
LongitCornerRebarSize = RebarSize;
ret2 = G.API.PropFrame.SetRebarColumn(...
Name, MatPropLong, MatPropConfine, Pattern, ...
ConfineType, Cover, NumberCBars, ...
NumberR3Bars, NumberR2Bars, RebarSize, ...
TieSize, TieSpacingLongit, Number2DirTieBars,...
Number3DirTieBars, ToBeDesigned);
if any([ret1,ret2])
disp("Error in setting " + Name)
end
end
```

Lampiran 16: beamRebarDesign1b

```
function [BeamRebarTable2, nBeamPenalty] =
beamRebarDesign1b(G,BeamRebarTable)
%% Reassign to G
fyt = G.Constants.fyt;
fy = G.Constants.fy;
TableKey = "Concrete Beam Design Summary - " + G.Constants.DesignCode;
BeamReinforcement=getTableDisplay(G.API.DatabaseTables,TableKey,"01_SM
F");
TableKey = "Frame Section Property Definitions - Concrete
Rectangular";
FrameSection = getTableDisplay(G.API.DatabaseTables, TableKey);
BeamSection = FrameSection(startsWith(FrameSection.('Name'),"G"),:);
%% Assign locations to stations
for i = 1:height(BeamReinforcement)
MaxSta=max(BeamReinforcement.('Station')(BeamReinforcement.('UniqueNam
e') == BeamReinforcement.('UniqueName')(i)));
MinSta=min(BeamReinforcement.('Station')(BeamReinforcement.('UniqueNam
e') == BeamReinforcement.('UniqueName')(i)));
LnBeam = MaxSta - MinSta;
h = BeamSection.("Depth")(string(BeamSection.("Name")) ==
string(BeamReinforcement.("DesignSect")(i)));
L_0 = 2*h;
BeamReinforcement.("L0 from 2h")(i) = L_0;
BeamReinforcement.("Ln")(i) = LnBeam;
if BeamReinforcement.Station(i)-MinSta <= L_0
BeamReinforcement.("Trv. Location")(i) = cellstr('End-I');
elseif BeamReinforcement.Station(i) >= MaxSta - L_0
BeamReinforcement.("Trv. Location")(i) =            cellstr('End-J');
else
BeamReinforcement.("Trv. Location")(i) = cellstr('Mid');
end
end
%% Obtain forces
% Set load cases
Sds = G.Constants.Sds;
LoadCases = NET.createArray("System.String",5);
```

Lampiran 16: beamRebarDesign1b (Lanjutan)

```
retSetLoadCases
=G.API.DatabaseTables.SetLoadCasesSelectedForDisplay(LoadCases);
LoadCases(1) = "DEAD";
LoadCases(2) = "LIVE";
LoadCases(3) = "GEMPAX";
LoadCases(4) = "GEMPAY";
retSetLoadCases=G.API.DatabaseTables.SetLoadCasesSelectedForDisplay(Lo
adCases);
ForcesTable = getTableDisplay(G.API.DatabaseTables,"Element Forces -
Beams","01_SMF");
ForcesTable1 = ForcesTable(ForcesTable.("Output Case")=="DEAD",:);
ForcesTable2 = ForcesTable(ForcesTable.("Output Case")=="LIVE",:);
ForcesTable3 = ForcesTable(ForcesTable.("Output Case")=="GEMPAX",:);
ForcesTable4 = ForcesTable(ForcesTable.("Output Case")=="GEMPAY",:);
% 0.5 L (D, L, GEMPAX, GEMPAY)
CombinationsMultiplier = [1.4, 0, 0, 0;...
                          1.2, 1.6, 0, 0;...
                          1.2 + 0.2*Sds, 0.5, 1.3, 0;...
                          1.2 + 0.2*Sds, 0.5, 0, 1.3;...
                          0.9 - 0.2*Sds, 0, 1.3, 0;
                          0.9 - 0.2*Sds, 0, 0, 1.3];...
%% Separate short and long beams
cond1 = BeamReinforcement.("L0 from 2h").*2 < BeamReinforcement.Ln;
%Ln > 4h condition
NormalBeamReinforcement = BeamReinforcement(cond1,:);
ShortBeamReinforcement = BeamReinforcement(~cond1,:);
%% Element forces
ForcesMatrix = zeros([height(ForcesTable1),6,4]);
ForcesMatrix(:,:,1)=table2array(ForcesTable1(:,["P","V2","V3","T","M2"
,"M3"]));
ForcesMatrix(:,:,2)=table2array(ForcesTable2(:,["P","V2","V3","T","M2"
,"M3"]));
ForcesMatrix(:,:,3)=table2array(ForcesTable3(:,["P","V2","V3","T","M2"
,"M3"]));
ForcesMatrix(:,:,4)=table2array(ForcesTable4(:,["P","V2","V3","T","M2"
,"M3"]));
%% Superimpose tables
VTables = ForcesMatrix(:,2,1:4);
```

Lampiran 16: beamRebarDesign1b (Lanjutan)

```
VTables = reshape(VTables,[height(VTables),4]);
VTables = VTables(~cond1,:);
%% Beam size tables
BeamSectionSizeTable =
BeamSection(:,["Name","Material","Depth","Width"]);
BeamSectionSizeTable=innerjoin(BeamSectionSizeTable,BeamRebarTable(:,[
"Name","RealEffDepthMinimum","AsProv"]));
BeamSectionSizeTable=renamevars(BeamSectionSizeTable,"Name","DesignSec
t");
%BeamSectionSizeTable.
phi = 0.6;
ShortBeamReinforcement =
innerjoin(ShortBeamReinforcement,BeamSectionSizeTable);
% Remove longitudinal details
% Vc calculation
a=ShortBeamReinforcement.AsProv.*fy*1.25./(0.85*G.Constants.fcBeam.*Sh
ortBeamReinforcement.Width);
Mpr=ShortBeamReinforcement.AsProv.*fy.*1.25.*(ShortBeamReinforcement.D
epth-ShortBeamReinforcement.RealEffDepthMinimum-a./2)*1000; %to kNm
Vp1 = (Mpr(:,1)+Mpr(:,6))./ShortBeamReinforcement.Ln;
Vp2 = (Mpr(:,3)+Mpr(:,4))./ShortBeamReinforcement.Ln;
Vp = max([Vp1, Vp2],[],2);
Vu = abs(VTables)*transpose(CombinationsMultiplier(3:6,:));
Vg = (VTables(:,1:2))*transpose(CombinationsMultiplier(4:5,1:2));
Ve = max(abs(Vg),[],2)+Vp;
% ShortBeamReinforcement.Vp = Vp;
ShortBeamReinforcement.RealEffDepthMinimum =
max(ShortBeamReinforcement.RealEffDepthMinimum,[],2);
Vc=phi*ShortBeamReinforcement.Width.*(ShortBeamReinforcement.Depth-
ShortBeamReinforcement.RealEffDepthMinimum).* ... % bw * d
sqrt(G.Constants.fcBeam).*0.17.*1000; %to kN
maxVc = Vc.*0.83./0.17; %from 0.17 to 0.17 + 0.66
VDesign = max([Ve,Vu],[],2);
ShortBeamShearPenalty = sum(VDesign > maxVc);   % simplified penalty,
torsion ignored
%% Get Av/s for short beams
cond2 = VDesign <= Vc./2*phi;
```

Lampiran 16: beamRebarDesign1b (Lanjutan)

```
cond3 = (VDesign <= maxVc) & ~cond2;
ShortBeamVRebar=(VDesign-Vc)./(phi*fyt*(ShortBeamReinforcement.Depth-
ShortBeamReinforcement.RealEffDepthMinimum));
ShortBeamVRebarMin =
max(0.062*sqrt(G.Constants.fcBeam),0.35).*ShortBeamReinforcement.Width
.*(ShortBeamReinforcement.Depth-
ShortBeamReinforcement.RealEffDepthMinimum);
ShortBeamVRebarUse = max([ShortBeamVRebar, ShortBeamVRebarMin],[],2);
ShortBeamReinforcement.AtTrnTotal = ShortBeamVRebarUse./1000;
NewBeamReinforcement=[NormalBeamReinforcement;ShortBeamReinforcement(:
,1:end-5)];
%% Assign rebar values to each beam
BeamRebarTable=join(BeamRebarTable,BeamSection(:,["Name","Depth","Widt
h"]));
% Matrix is based off BeamRebarTable
AtTrnTotalMatrix = zeros(height(BeamRebarTable),3);
LnColumn = zeros(height(BeamRebarTable),1);
for i = 1:height(BeamRebarTable)
AtTrnTotalMatrix(i,1)=max(NewBeamReinforcement.("AtTrnTotal")((string(
NewBeamReinforcement.('DesignSect')) ==
string(BeamRebarTable.('Name')(i))) & (NewBeamReinforcement.('Trv.
Location') == "End-I")));
try
AtTrnTotalMatrix(i,2) =
max(NewBeamReinforcement.("AtTrnTotal")((string(NewBeamReinforcement.(
'DesignSect')) == string(BeamRebarTable.('Name')(i))) &
(NewBeamReinforcement.('Trv. Location') == "Mid")));
catch
AtTrnTotalMatrix(i,2) = 0;
end
try
AtTrnTotalMatrix(i,3)=max(NewBeamReinforcement.("AtTrnTotal")((string(
NewBeamReinforcement.('DesignSect')) ==
string(BeamRebarTable.('Name')(i))) & (NewBeamReinforcement.('Trv.
Location') == "End-J")));
catch
AtTrnTotalMatrix(i,3) = 0;
end
```

Lampiran 16: beamRebarDesign1b (Lanjutan)

```
LnColumn(i,1)=max(NewBeamReinforcement.("Ln")((string(NewBeamReinforce
ment.('DesignSect')) == string(BeamRebarTable.('Name')(i)))));
end
ConcreteCover = G.Constants.ConcreteCover;
MinimumSpacingParallel = G.Constants.MinimumSpacingParallel;
MinimumSpacingVertical = 0.025;
MultipleSpacingTrv = G.Constants.MultipleSpacingTrv;
AtColumn = (BeamRebarTable.dtMinimum./1000).^2./4.*pi();
MaxPossibleRebarPerLayer = floor((BeamRebarTable.("Width")-
2*ConcreteCover-
2.*BeamRebarTable.dtMinimum./1000+MinimumSpacingParallel)./(BeamRebarT
able.dbMinimum./1000+MinimumSpacingParallel));
MaxActualRebarPerLayer = max(BeamRebarTable.nReinMinimum,[],2);
MaxRebarPerLayer =
min([MaxPossibleRebarPerLayer,MaxActualRebarPerLayer],[],2);
% Real max distance
MaxDistance1=min([(BeamRebarTable.Depth-
max(BeamRebarTable.RealEffDepthMinimum,[],2))./4,
0.15*ones(height(BeamRebarTable),1),
6.*BeamRebarTable.dbMinimum./1000],[],2);
MaxDistance2=(BeamRebarTable.Depth-
max(BeamRebarTable.RealEffDepthMinimum,[],2))./2;
% Practical max distance
MaxDistance1=floor(MaxDistance1./MultipleSpacingTrv).*MultipleSpacingT
rv;
MaxDistance2=floor(MaxDistance2./MultipleSpacingTrv).*MultipleSpacingT
rv;
MaxDistance = [MaxDistance1, MaxDistance2, MaxDistance1];
MaxNumberLegsRequired = ceil(AtTrnTotalMatrix.*MaxDistance./AtColumn);
MinNumberLegsRequired =
ceil(AtTrnTotalMatrix.*G.Constants.MinimumSpacingTrv./AtColumn);
%% Using minimum number of legs:
MinNumberLegsRequired =
(MinNumberLegsRequired>0).*max(2,MinNumberLegsRequired);
TrvSpacing = MinNumberLegsRequired.*AtColumn./(AtTrnTotalMatrix);
TrvSpacing=floor(TrvSpacing./MultipleSpacingTrv).*MultipleSpacingTrv;
%% Insert to BeamRebarTable
BeamRebarTable.nTrvLegs = MinNumberLegsRequired;
```

Lampiran 16: beamRebarDesign1b (Lanjutan)

```
BeamRebarTable.TrvSpacing = TrvSpacing;
BeamRebarTable.LnColumn = LnColumn;
%% Temp
BeamRebarTable2 = BeamRebarTable;
%% Penalty constraints
BeamShearPenalty =
height(NormalBeamReinforcement.ErrMsg(startsWith(NormalBeamReinforceme
nt.ErrMsg,"Shear stress")));
BeamCompressionPenalty =
height(BeamReinforcement.ErrMsg(startsWith(BeamReinforcement.ErrMsg,"B
eam concrete compression")));
BeamMomentPenalty =
height(BeamReinforcement.ErrMsg(startsWith(BeamReinforcement.ErrMsg,"R
einforcing required exceeds")));
nBeamPenalty = BeamShearPenalty + BeamCompressionPenalty +
BeamMomentPenalty + ShortBeamShearPenalty;
```

Lampiran 17: constraint_column

```
function [totalColPenalty, sixpercentPenalty, threepercentPenalty,
colShearPenalty, colBucklingPenalty] = constraint_column(G)
% This function checks the failure of "To Be Designed" columns.
% Access DatabaseTables and DesignConcrete interfaces
DatabaseTables = G.API.DatabaseTables;
DesignConcrete = G.API.DesignConcrete;
% Gets column design summary table
CodeName = "";
[ret, CodeName] = DesignConcrete.GetCode(CodeName);
CodeName = string(CodeName);
TableKey = "Concrete Column Design Summary - " + CodeName;
columnTable = getTableDisplay(DatabaseTables,TableKey,"01_SMF");
% Checks for 6% As and 3% As
sixpercentPenalty = sum(columnTable.As./columnTable.AsMin > 6);
%threepercentPenalty = sum(columnTable.As./columnTable.AsMin > 3);
PMMPenalty = sum(columnTable.("PMMRatio")>G.Constants.MaxPMMRatio);
% Checks for error messages containing shear and buckling failures
colShearPenalty =
height(columnTable.ErrMsg(startsWith(columnTable.ErrMsg,"Shear
stress")));
colBucklingPenalty =
height(columnTable.ErrMsg(startsWith(columnTable.ErrMsg,"Column
factored axial")));
totalColPenalty = sixpercentPenalty + colShearPenalty +
colBucklingPenalty + PMMPenalty;
```

Lampiran 18: constraint_BCC_JS

```
function [totalJointPenalty, BCCPenalty, jointPenalty] =
constraint_BCC_JS(G)
% This function checks for joints that exceed the maximum BCC and JS.
% [BCCPenalty, jointPenalty] = constraint_BCC_JS(G)
% Access DatabaseTables and DesignConcrete interfaces
DatabaseTables = G.API.DatabaseTables;
DesignConcrete = G.API.DesignConcrete;
% Max BCC and JS ratio before given penalty // Can be edited
MaxBCCRatio = G.Constants.MaxBCCRatio;
MaxJSRatio = G.Constants.MaxJSRatio;
% Gets joint design summary table
TableKey = "Concrete Joint Design Summary - " +
G.Constants.DesignCode;
JointCheck = getTableDisplay(DatabaseTables,TableKey,"01_SMF");
TopJointCheck =
getTableDisplay(DatabaseTables,TableKey,"04_TOPCOLUMNS");
[~,idx2]=ismember(JointCheck.DesignSect,TopJointCheck.DesignSect);
NoBCCJointCheck = JointCheck(~idx2,:);
% Calculated the amount of BCC and JS penalties
BCCPenalty = sum([NoBCCJointCheck.BCMajRatio > MaxBCCRatio;
NoBCCJointCheck.BCMinRatio > MaxBCCRatio]);
jointPenalty = sum([JointCheck.JSMajRatio > MaxJSRatio;
JointCheck.JSMinRatio > MaxJSRatio]);
totalJointPenalty = BCCPenalty + jointPenalty;
```

Lampiran 19: constraint_drift

```
function driftPenalty = constraint_drift(G)
% This function checks the drift limits by checking the drifts
reported in
% the "Story Drifts" table output.
% driftPenalty = constraint_drift(G)
DatabaseTables = G.API.DatabaseTables;
% Input importance factor, Ie and maximum allowable story drift
Ie = G.Constants.Ie;
Cd = G.Constants.Cd;
maxDrift = G.Constants.maxDrift;
% - Editable // Input names of earthquake load cases from ETABS
accordingly
EQX_Name = "GEMPAX";
EQY_Name = "GEMPAY";
% Sets the load case for output
LoadCaseList = NET.createArray('System.String',3);
LoadCaseList(1) = EQX_Name;
LoadCaseList(2) = EQY_Name;
retSetLoadCases=DatabaseTables.SetLoadCasesSelectedForDisplay(LoadCase
List);
% Get story drifts table from load cases set for output
TableKey = "Story Drifts";
DriftTable = getTableDisplay(DatabaseTables,TableKey);
% Gets story drifts for x and y direction
DriftX = DriftTable.("Drift")((string(DriftTable.Direction) == ('X'))
& (DriftTable.("Output Case") == EQX_Name));
DriftY = DriftTable.("Drift")((string(DriftTable.Direction) == ('Y'))
& (DriftTable.("Output Case") == EQY_Name));
% Calculate the amount of stories that exceeds the allowable drift
limits
driftPenalty = sum([DriftX*Cd/Ie>maxDrift; DriftY*Cd/Ie>maxDrift]);
```

Lampiran 20: constraint_beam

```
function [totalBeamPenalty, beamShearPenalty, beamCompressionPenalty,
beamMomentPenalty] = constraint_beam(G)
% This function checks for beams that exceeds limits given by ETABS
% Concrete Design.
% [beamShearPenalty, beamCompressionPenalty, beamMomentPenalty] =
constraint_beam(G)
% Access DatabaseTables and DesignConcrete interfaces
DatabaseTables = G.API.DatabaseTables;
DesignConcrete = G.API.DesignConcrete;
% Gets beam design summary table
CodeName = "";
[ret, CodeName] = DesignConcrete.GetCode(CodeName);
CodeName = string(CodeName);
TableKey = "Concrete Beam Design Summary - " + CodeName;
BeamCheck = getTableDisplay(DatabaseTables,TableKey,"01_SMF");
% Checks beam failures
beamShearPenalty =
height(BeamCheck.ErrMsg(startsWith(BeamCheck.ErrMsg,"Shear stress")));
beamCompressionPenalty =
height(BeamCheck.ErrMsg(startsWith(BeamCheck.ErrMsg,"Beam concrete
compression")));
beamMomentPenalty =
height(BeamCheck.ErrMsg(startsWith(BeamCheck.ErrMsg,"Reinforcing
required exceeds")));
totalBeamPenalty = beamShearPenalty + beamCompressionPenalty +
beamMomentPenalty;
```

Lampiran 21: getDimLowerBound

```
function [output, ret, hBeamLowerBound, bBeamLowerBound,
hColLowerBound, bColLowerBound] = getDimLowerBound(DatabaseTables)
% This function calculates the lower bound of special moment frame
sections
% based on dimensions constraints. The lower bound of beams is also
% determined by the length.
% Multiplier from 9.3.1.1
Multiplier = 16;
% Gets frame sections and its lengths
TableKey = "Frame Assignments - Summary";
TableOutput = getTableDisplay(DatabaseTables, TableKey, "01_SMF");
% Sorts the table and separate beams, columns, and everything else
SortedTableOutput = sortrows(TableOutput,'Analysis Section','ascend');
BeamTableOutput=SortedTableOutput(startsWith(SortedTableOutput.('Analy
sis Section'),"G"),:);
ColTableOutput=SortedTableOutput(startsWith(SortedTableOutput.('Analys
is Section'),"K"),:);
NoCols=sum(~startsWith(SortedTableOutput.('Analysis Section'),"G"));
% Get lower bound of beam depths by checking its length and width by
% checking its minimum height. Numbers are rounded to the nearest 50
mm.
hBeamLowerBound=ceil(BeamTableOutput.('Length')./Multiplier./0.05)*0.0
5;
bBeamLowerBound = ceil(max(hBeamLowerBound*0.3,0.250)/0.05)*0.05;
% Column dimensions use 300mm as lower bound
hColLowerBound = ones(1,NoCols)*0.3;
bColLowerBound = ones(1,NoCols)*0.3;
% Merge results into one array
output = [transpose(hBeamLowerBound) transpose(bBeamLowerBound)
hColLowerBound bColLowerBound];
```

Lampiran 22: getTableDisplay

```
function
[TableOutput,csvFilePath]=getTableDisplay(DatabaseTables,TableKey,Grou
p)
% This function exports ETABS tables to csv files to be then imported
to% MATLAB workspace as a table.
% Sets default group to All

if (~exist('Group', 'var'))
Group = "";
end
% Create csv file of table and reads it
string_array = NET.createArray('System.String',2);
FilePath = fileparts(pwd);
FieldKeyList = string_array;
csvFilePath = FilePath+"\csv\"+TableKey+".csv";
ret =
DatabaseTables.GetTableForDisplayCSVFile(TableKey,FieldKeyList,Group,1
,csvFilePath);
TableOutput = readtable(csvFilePath,'PreserveVariableNames',true);
delete(csvFilePath)
```

Lampiran 23: Algoritma Optimasi SOS Assisted

```
clear;clc;close all;

tic;
% Connect MATLAB to ETABS
parentFolder = fileparts(pwd);
listing = dir(parentFolder);
for iFiles = 1:length(listing)
if listing(iFiles).isdir
addpath(parentFolder + "\" + listing(iFiles).name)
end
end
G = globalVar;
%% Added ETABSObject to all fobj
d1=datetime;
%initial_sections=readmatrix("InitialSections.csv");
initial_population = readmatrix("Copy_of_Population40Mixedv1.csv");
initial_sections = initial_population(1,:);
% editable
array_1 = 1:G.Constants.nBeams;
array_2 =
(G.Constants.nBeams*2+1):((G.Constants.nBeams+G.Constants.nColumns)*2)
;
lb_1=getDimLowerBound(G.API.DatabaseTables);
lb_1=lb_1([array_1,array_2]); %changed to only
lb_1 = max(lb_1,0.5);
lb = lb_1;
%ub_1=ones(1,length(lb))*1.5; % highly consider changing this
ub_1=[ones(1,G.Constants.nBeams)*1,
ones(1,G.Constants.nColumns*2)*1.2];
ub = ub_1; % constrainted upper bounds
initial_population = initial_population(:,[array_1,array_2]);
%bound the population
for i=1:height(initial_population)
initial_population(i,:) = bound(initial_population(i,:),ub,lb); %
end
G.Bounds.LB = lb; G.Bounds.UB = ub;
fobj=@obj_func3_assisted;
iteration=5;
```

Lampiran 23: Algoritma Optimasi SOS Assisted (Lanjutan)

```
ecosize=height(initial_population);
n=length(lb);
maxFE=iteration*ecosize*4; % multiply by 4
FE=0;
%eco = zeros(ecosize,n);
% --- Ecosystem initialization
fitnessNew1=zeros(ecosize,2);
ecoNew1=zeros(ecosize,n);
fitnessNew2=zeros(ecosize,2);
ecoNew2=zeros(ecosize,n);
fitnessNew3=zeros(ecosize,2);
ecoNew3=zeros(ecosize,n);
fitnessNew4=zeros(ecosize,2);
ecoNew4=zeros(ecosize,n);
fitness = zeros(ecosize,2);
fitnessmatrix = zeros(ecosize,iteration+1);
fitnessmatrix(:,1) = fitness(:,1);
iter=0;
% -- Initial population;
eco = initial_population;
ecomatrix = zeros([size(eco), iteration+1]);
%% Ecosystem Initialization
for i=1:ecosize
[fitness(i,1),fitness(i,2)]=fobj(G,eco(i,:));
end
fmax=max(fitness(:,1).*(fitness(:,2)==0));
for i=1:ecosize
if fitness(i,2)>0
fitness(i,1)=fmax+fitness(i,2);
end
end
% sort & update the best initial organism
[fitness, l]=sortrows(fitness,1); eco=eco(l,:);
[fitness, m]=sortrows(fitness,2); eco=eco(m,:);
[bestFitness]=fitness(1,1); bestOrganism=eco(1,:);
ii=1;
%% Main Looping
while FE<maxFE
```

Lampiran 23: Algoritma Optimasi SOS Assisted (Lanjutan)

```
for i=1:ecosize % Organisms' Looping
% Update the best Organism
[~,idx]=min(fitness(:,1)); bestOrganism=eco(idx,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mutualism Phase
% Choose organism j randomly other than organism i
j=i;
while i==j
seed=randperm(ecosize);
j=seed(1);
end
% Determine Mutual Vector & Beneficial Factor
mutualVector=mean([eco(i,:); eco(j,:)]);
BF1=round(1+rand); BF2=round(1+rand);
% Calculate new solution after Mutualism Phase
ecoNew1(i,:)=eco(i,:)+rand(1,n).*(bestOrganism-BF1.*mutualVector);
ecoNew2(j,:)=eco(j,:)+rand(1,n).*(bestOrganism-BF2.*mutualVector);
ecoNew1(i,:)=bound(ecoNew1(i,:),ub,lb);
ecoNew2(j,:)=bound(ecoNew2(j,:),ub,lb);
% Evaluate the fitness of the new solution
[fitnessNew1(i,1),fitnessNew1(i,2)]=fobj(G,ecoNew1(i,:));
[fitnessNew2(j,1),fitnessNew2(j,2)]=fobj(G,ecoNew2(j,:));
if fitnessNew1(i,2)==0
elseif fitnessNew1(i,2)>0
fitnessNew1(i,1)=fmax+fitnessNew1(i,2);
end
if fitnessNew2(j,2)==0
elseif fitnessNew2(j,2)>0
fitnessNew2(j,1)=fmax+fitnessNew2(j,2);
end
if fitnessNew1(i,2)==fitness(i,2)
if fitnessNew1(i,1)<fitness(i,1)
fitness(i,:)=fitnessNew1(i,:);
eco(i,:)=ecoNew1(i,:);
end
elseif fitnessNew1(i,2)<fitness(i,2)
fitness(i,:)=fitnessNew1(i,:);
eco(i,:)=ecoNew1(i,:);
```

Lampiran 23: Algoritma Optimasi SOS Assisted (Lanjutan)

```
end
if fitnessNew2(j,2)==fitness(j,2)
if fitnessNew2(j,1)<fitness(j,1)
fitness(j,:)=fitnessNew2(j,:);
eco(j,:)=ecoNew2(j,:);
end
elseif fitnessNew2(j,2)<fitness(j,2)
fitness(j,:)=fitnessNew2(j,:);
eco(j,:)=ecoNew2(j,:);
end
% Accept the new solution if the fitness is better
% Increase the number of function evaluation counter
FE=FE+2;
% End of Mutualism Phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Commensialism Phase
% Choose organism j randomly other than organism i
j=i;
while i==j
seed=randperm(ecosize);
j=seed(1);
end
ecoNew3(i,:)=eco(i,:)+(rand(1,n)*2-1).*(bestOrganism-eco(j,:));
ecoNew3(i,:)=bound(ecoNew3(i,:),ub,lb);
[fitnessNew3(i,1),fitnessNew3(i,2)]=fobj(G,ecoNew3(i,:));
if fitnessNew3(i,2)==0
elseif fitnessNew3(i,2)>0
fitnessNew3(i,1)=fmax+fitnessNew3(i,2);
end
% Accept the new solution if the fitness is better
eco(i,:)]=compareTwoSolution(fitnessNew3,fitness(i,:),ecoNew3,eco(i,:)
);
if fitnessNew3(i,2)==fitness(i,2)
if fitnessNew3(i,1)<fitness(i,1)
fitness(i,:)=fitnessNew3(i,:);
eco(i,:)=ecoNew3(i,:);
end
elseif fitnessNew3(i,2)<fitness(i,2)
```

Lampiran 23: Algoritma Optimasi SOS Assisted (Lanjutan)

```
fitness(i,:)=fitnessNew3(i,:);
 eco(i,:)=ecoNew3(i,:);
end
% Increase the number of function evaluation counter
FE=FE+1;
% End of Commensalism Phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parasitism Phase
% Choose organism j randomly other than organism i
j=i;
while i==j
seed=randperm(ecosize);
j=seed(1);
end
% Determine Parasite Vector & Calculate the fitness
parasiteVector=eco(i,:);
seed=randperm(n);
pick=seed(1:ceil(rand*n));  % select random dimension
parasiteVector(:,pick)=rand.*(ub(pick)-lb(pick))+lb(pick);
ecoNew4(i,:)=parasiteVector;
% Evaluate the fitness of the new solution
[fitnessNew4(i,1),fitnessNew4(i,2)]=fobj(G,ecoNew4(i,:));
if fitnessNew4(i,2)==0
elseif fitnessNew4(i,2)>0
fitnessNew4(i,1)=fmax+fitnessNew4(i,2);
end
% Kill organism j and replace it with the parasite
% if the fitness is lower than the parasite
%[fitness(i,:)
eco(i,:)]=compareTwoSolution(fitnessParasite,fitness(i,:),parasiteVect
or,eco(i,:));
if fitnessNew4(i,2)==fitness(j,2)
if fitnessNew4(i,1)<fitness(j,1)
 fitness(j,:)=fitnessNew4(i,:);
eco(j,:)=ecoNew4(i,:);
end
elseif fitnessNew4(i,2)<fitness(j,2)
fitness(j,:)=fitnessNew4(i,:);
```

Lampiran 23: Algoritma Optimasi SOS Assisted (Lanjutan)

```
eco(j,:)=ecoNew4(i,:);
end
% Increase the number of function evaluation counter
FE=FE+1;
% End of Parasitism Phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end % End of Organisms' Looping
fmax=max(fitness(:,1).*(fitness(:,2)==0));
for i=1:ecosize
if fitness(i,2)>0
fitness(i,1)=fmax+fitness(i,2);
end
end
% sort and update the best organism
[fitness, l]=sortrows(fitness,1);eco=eco(l,:);
[fitness, m]=sortrows(fitness,2); eco=eco(m,:);
bestOrganism=eco(1,:); bestFitness=fitness(1,1);
fitnessmatrix(:,ii) = fitness(:,1);
ecomatrix(:,:,ii) = eco;
ii=ii+1;
end % End of Main Looping
%% Display the result
disp(['Funnum: ', num2str(funnum)])
disp(['fitness: ', num2str(bestFitness)])
disp(['Solution: ', num2str(bestOrganism)])
% save('parameter.mat','ecosize','MR','CR','PR','AR')
%% Post-processing
d2 = datetime;
runtime = d2-d1; disp(runtime);
writematrix([fitness,eco],"Results -
"+string(datetime('now','Format','yyyy-MM-dd_HH_mm_ss'))+".csv")
casefilename = 'CaseSOS2Assisted-
'+string(datetime('now','Format','yyyy-MM-dd_HH_mm_ss'))+'.mat';
save(casefilename)
%% Subfunctions
% Check the boundary limit
function a=bound(a,ub,lb)
a(a>ub)=ub(a>ub); a(a<lb)=lb(a<lb);
```

Lampiran 23: Algoritma Optimasi SOS Assisted (Lanjutan)

```
end
function fitness=fitnessEval(fitness,eco,funnum,fmax,data)
if fitness(1,2)==0
fitness(1,1)=fobj(eco,funnum,data);
elseif fitness(1,2)>0
fitness(1,1)=fmax+fitness(1,2);
end
end
function [fitness
eco]=compareTwoSolution(fitnessNew,fitness,ecoNew,eco)
if fitnessNew(1,2)==fitness(1,2)
if fitnessNew(1,1)<fitness(1,1)
fitness=fitnessNew;
eco=ecoNew;
end
elseif fitnessNew(1,2)<fitness(1,2)
fitness=fitnessNew;
eco=ecoNew;
end
end
```

Lampiran 24: obj_func3_assisted

```
function [Cost, nPenalty, iBCCDesign, TableList] =
obj_func3_assisted(G,x)
% Main objective function
% Cost = obj_func3(G,x)
% Where:
% G is the global structure
% x is an array consisting of beam and column dimensions
% [T2Beam, T3Beam, T2Column, T3Column]
d1 = datetime;
%% Get number of beams and columns from global
nBeams = G.Constants.nBeams;
nColumns = G.Constants.nColumns;
%% Get price
conc_strength = "fc " +
G.Tables.MaterialPropertiesConcreteData.Fc/1000;
% conc_strength = "fc 40"; % User input
conc_price = G.PriceList.Harga(find(G.PriceList.("Material") ==
conc_strength));
steel_price = G.PriceList.Harga(find(G.PriceList.("Material") ==
"Baja"));
%% Separate x into beams and columns
% bound first
%x = bound(x,G.Bounds.UB,G.Bounds.LB);
% Rounds up each dimension to the nearest 50 cm
x = round(x./0.05).*0.05;
T2Beam = round(x(1:nBeams)./0.05)*0.05;
T3Beam = ceil(T2Beam.*0.6./0.05)*0.05; % to 0.6 instead
T2Column = round(x((nBeams+1):(nBeams+nColumns))./0.05)*0.05;
T3Column =
round(x((nBeams+nColumns+1):(nBeams+nColumns*2))./0.05)*0.05;
%% Dimensions constraint penalty --> Force into constraints
[~,~,~,x] = constraint_dimension(T2Beam,T3Beam,T2Column,T3Column,G);
x = round(x./0.05).*0.05;
end
%% Reset reinforcement
retInitialize = resetSections2(G,x);
%% 60-90+ seconds
if retInitialize == 1 % returns function if fails
```

Lampiran 24: obj_func3_assisted (Lanjutan)

```
Cost = 0; nPenalty = Inf;
return
end
%% Run the structural analysis
for i = 1:2
retAnalysis = G.API.Analyze.RunAnalysis; %% 30+ seconds
%% Scale response spectrum base shear // add torsion here. 15+ secs
RS_ScaleSF(G);
RS_EccScale(G);
%% Run concrete design on special moment beams and top columns only.
5+ seconds
retDeselectAll =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retSelectBeams =
G.API.FrameObj.SetSelected("02_SMBEAMS",true,ETABSv1.eItemType.Group);
%Data might be corrupted/be inconsistent if done this way
retStartDesign = G.API.DesignConcrete.StartDesign;
%% Make new beam sections
beam_checking_script(G);
end
%% Run the structural analysis and scale response spectrum again
retAnalysis = G.API.Analyze.RunAnalysis;
RS_ScaleSF(G);
RS_EccScale(G);
retDeselectAll =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retSelectBeams =
G.API.FrameObj.SetSelected("02_SMBEAMS",true,ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
BeamRebarTable = beamRebarDesign1a(G);
%% Reinput reinforcement to ETABS.
retSetRebar1 = setRebar1_1(G,BeamRebarTable);
%% Rerun analysis and concrete design.
retAnalysis = G.API.Analyze.RunAnalysis;
% Top column design
tic
retChangeBCCPreference = G.API.ACI318_14.SetPreference(4,1); % turn
off design for BCC
```

Lampiran 24: obj_func3_assisted (Lanjutan)

```
retSelectFrames =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retSelectBeams =
G.API.FrameObj.SetSelected("02_SMBEAMS",true,ETABSv1.eItemType.Group);
retDeselectTopColumns =
G.API.FrameObj.SetSelected("04_TOPCOLUMNS",true,ETABSv1.eItemType.Grou
p);
retStartDesign = G.API.DesignConcrete.StartDesign;
TopColumnReinTable = columnRebarDesign1a(G);
% All column design
retChangeBCCPreference = G.API.ACI318_14.SetPreference(4,2); % turn on
design for BCC
retSelectFrames =
G.API.FrameObj.SetSelected("All",true,ETABSv1.eItemType.Group);
retDeselectTopColumns =
G.API.FrameObj.SetSelected("04_TOPCOLUMNS",false,ETABSv1.eItemType.Gro
up);
retStartDesign = G.API.DesignConcrete.StartDesign;
ColumnRebarTable = columnRebarDesign1b(G,TopColumnReinTable);
retSetRebar2 = setRebar2_1(G,ColumnRebarTable);
toc
%% Set rebar 3 column design
iBCCDesign = 0;
RedesignFlag = true;
maxiBCCDesign = 1;
while RedesignFlag == true && iBCCDesign <= maxiBCCDesign
iBCCDesign = iBCCDesign + 1;
retAnalysis = G.API.Analyze.RunAnalysis;
retDeselectAll =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
[~, RedesignFlag] = columnAsDesign(G);
end
if RedesignFlag == true
maxiii = 2;
else
maxiii  = 1;
end
```

Lampiran 24: obj_func3_assisted (Lanjutan)

```
for iii=1:maxiii %do this twice
retAnalysis = G.API.Analyze.RunAnalysis;
retDeselectAll =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
%Column shear design
ColumnTable = columnAvDesign(G);
setRebar3_1(G,ColumnTable);
% Final concrete design
retAnalysis = G.API.Analyze.RunAnalysis;
retDeselectAll =
G.API.FrameObj.SetSelected("All",false,ETABSv1.eItemType.Group);
retStartDesign = G.API.DesignConcrete.StartDesign;
ColumnTable = columnAvDesign2(G);
%% Check for constraints
%Beam shear design
[BeamRebarTable, nBeamPenalty] = beamRebarDesign1b(G,BeamRebarTable);
nColumnPenalty = constraint_column(G);
nJointPenalty = constraint_BCC_JS(G);
nDriftPenalty = constraint_drift(G);
%% Calculate steel cost
%% Reinforcement tables
FrameSum = G.Tables.FrameSum;
FrameSum = FrameSum(:,["Analysis Section","Length"]);
FrameSum = renamevars(FrameSum,'Analysis Section','Name');
BeamRebarTable = innerjoin(BeamRebarTable,FrameSum);
ColumnTable = innerjoin(ColumnTable,FrameSum);
ColumnTotalRebar = sum(ColumnTable.("As").*ColumnTable.("Length"));
TableList = struct;
TableList.BeamRebarTable = BeamRebarTable;
TableList.ColumnTable = ColumnTable;
Trv1(:,1) =
ceil(0.33.*BeamRebarTable.("LnBeam")./BeamRebarTable.TrvSpacing(:,1));
Trv1(:,2) =
ceil(0.34.*BeamRebarTable.("LnBeam")./BeamRebarTable.TrvSpacing(:,2));
Trv1(:,3) =
ceil(0.33.*BeamRebarTable.("LnBeam")./BeamRebarTable.TrvSpacing(:,3));
Trv1(isnan(Trv1))=0;
```

Lampiran 24: obj_func3_assisted (Lanjutan)

```
Trv1(:,4) = (BeamRebarTable.("Width")-
0.04*2)*2+((BeamRebarTable.("Depth")-
0.04*2).*BeamRebarTable.nTrvLegs(:,1));
Trv1(:,5) = (BeamRebarTable.("Width")-
0.04*2)*2+((BeamRebarTable.("Depth")-
0.04*2).*BeamRebarTable.nTrvLegs(:,2));
Trv1(:,6) = (BeamRebarTable.("Width")-
0.04*2)*2+((BeamRebarTable.("Depth")-
0.04*2).*BeamRebarTable.nTrvLegs(:,3));
Trv1(:,7) = Trv1(:,1).*Trv1(:,4);
Trv1(:,8) = Trv1(:,2).*Trv1(:,5);
Trv1(:,9) = Trv1(:,3).*Trv1(:,6);
Trv1(:,10) = Trv1(:,7)+Trv1(:,8)+Trv1(:,9);
Trv1(:,11) =
Trv1(:,10).*(0.25.*3.14./1000000.*BeamRebarTable.("dtMinimum").^2);
ShearBeam = sum(Trv1(:,11));
Trv2(:,1) = ceil(ColumnTable.("Length")./ColumnTable.("SpacingUse"));
Trv2(:,2) = ColumnTable.("Depth")-0.04*2;
Trv2(:,3) = ColumnTable.("Width")-0.04*2;
Trv2(:,4) =
((Trv2(:,2).*ColumnTable.("NumberOfLegsMaj")+Trv2(:,3).*ColumnTable.("
NumberOfLegsMin"))).*Trv2(:,1);
Trv2(:,5) =
Trv2(:,4).*0.25.*3.14.*(ColumnTable.("DiameterTransversal")./1000).^2;
ShearColumn = sum(Trv2(:,5));
TotalRebar = ColumnTotalRebar + BeamTotalRebar + ShearBeam +
ShearColumn;
CostConcrete = smf_weight(G)./23.56.*conc_price;
CostSteel = TotalRebar * 7850 * steel_price;
Cost = CostConcrete + CostSteel;
TableList.BeamPenalty = nBeamPenalty;
TableList.ColumnPenalty = nColumnPenalty;
TableList.JointPenalty = nJointPenalty;
TableList.DriftPenalty = nDriftPenalty;
d2 = datetime;
time = d2-d1;
disp(time)
```

Lampiran 25: bound

```
function a=bound(a,ub,lb)
a(a>ub)=ub(a>ub); a(a<lb)=lb(a<lb);
end
```

Lampiran 26: SOS *assisted*

Pada penelitian ini karena masih banyak elemen balok dan kolom yang diberikan tulangan sesuai Asmin. Peneliti menggunakan *engineering judgement* untuk membantu mempercepat proses optimasi yang dinamakan SOS *Assisted*. SOS *Assisted* dilakukan untuk menekan biaya struktur balok lebih rendah lagi. Hal ini dilakukan dengan melakukan pengecilan ukuran balok dalam *objective function*. Percobaan yang baru dilakukan dengan menjalankan SOS Assisted selama 5 iterasi menggunakan populasi awal dengan beberapa parameter yang ditentukan oleh peneliti. Perhitungan biaya, dan detail dari desain dengan SOS *assisted* dapat dilihat pada Gambar dan Tabel dibawah.

Parameter yang digunakan untuk metode SOS Assisted antara lain:

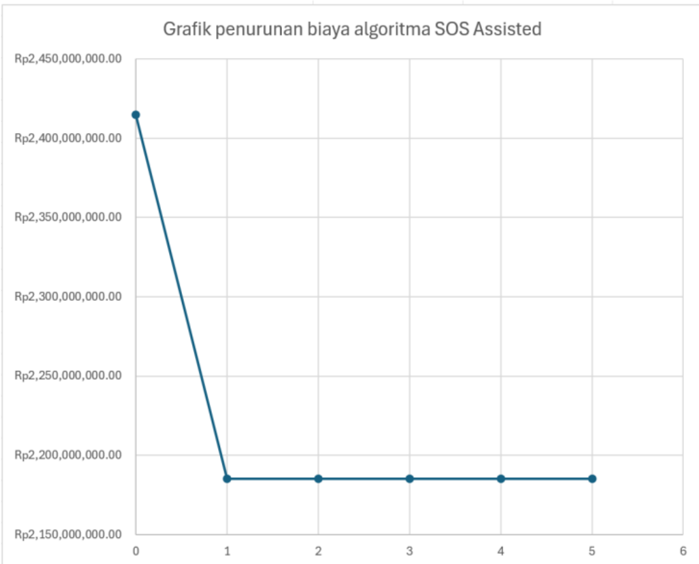
a. *Upper bound* kolom ditekan menjadi 1.2 m

b. *Upper bound* tinggi balok ditekan menjadi 1 m

c. *Upper bound* lebar balok ditekan menjadi 0.6 m

d. Rasio tinggi dan lebar balok di*bound* menjadi antara 2 dan 3

Populasi awal dari SOS *assisted* sudah memiliki biaya yang lebih rendah daripada hasil desain awal. Meskipun hanya ukuran balok yang diubah, pengaruhnya cukup besar terhadap biaya total dari biaya SRPMK. Parameter yang digunakan ditentukan dengan *engineering judgment*, dimana *engineer* dapat menyesuaikan parameter agar memiliki hasil yang lebih optimal.



Gambar Grafik penurunan biaya algoritma SOS *Assisted* studi kasus gedung 6 lantai

Lampiran 26: SOS *Assisted* (Lanjutan)

Tabel Perbandingan volume tulangan dan beton terhadap elemen balok dan kolom pada SOS assisted

Elemen Kolom	Jumlah	Harga Satuan (Rp)	Harga Total
Volume Tulangan Kolom (kg)	68.010	12.500	850.128.121
Volume Beton Kolom (fc' = 40 MPa, m^3)	318,997	1.120.000	357.277.709
Elemen Balok			
Volume Tulangan Balok (kg)	50.619	12.500	632.743.389
Volume Beton Balok (fc' = 40 MPa, m^3)	307,911	1.120.000	344.860.569
Harga Total (dalam rupiah)			**Rp 2.185.102.500**

Tabel Hasil optimasi SOS *assisted* elemen struktur kolom studi kasus

Lantai	Nama ID Elemen	Panjang Kolom (m)	Lebar Kolom (m)	Tinggi Kolom(m)	Tulangan Longitudinal	Tulangan Sengkang arah y	Tulangan Sengkang arah x
LT.1	K1	1,15	1,1	1,5	44D25	7D13-100	7D13-100
LT.1	K2	1,2	0,9	1,5	28D25	9D10-100	13D10-100
LT.1	K3	1	1,05	1,5	30D22	8D10-75	8D10-75
LT.1	K4	1,15	1,05	1,5	32D22	8D10-75	9D10-75
LT.1	K5	0,95	1	1,5	20D25	6D13-100	6D13-100
LT.1	K6	0,95	1,1	1,5	28D22	7D13-100	6D13-100
LT.1	K7	0,9	0,9	1,5	24D22	9D10-100	9D10-100
LT.1	K8	1,15	0,5	1,5	24D22	6D10-100	12D10-100
LT.1	K9	1	1	1,5	28D22	6D13-100	6D13-100
LT.1	K10	1,15	0,95	1,5	30D22	6D13-100	7D13-100
LT.1	K11	1,15	0,55	1,5	18D22	4D10-75	9D10-75
LT.1	K12	1,15	0,5	1,5	18D22	5D10-100	12D10-100
LT.1	K13	1,2	0,5	1,5	34D22	7D10-100	13D10-100
LT.1	K14-1	0,6	0,7	1,5	12D22	7D10-100	6D10-100
LT.1	K15-1	0,55	0,5	1,5	8D22	3D13-75	3D13-75

Tabel Hasil optimasi SOS *assisted* elemen struktur kolom studi kasus (Lanjutan)

Lantai	Nama ID Elemen	Panjang Kolom (m)	Lebar Kolom (m)	Tinggi Kolom(m)	Tulangan Longitudinal	Tulangan Sengkang arah y	Tulangan Sengkang arah x
LT.1	K16	0,95	0,9	1,5	28D22	9D10-100	10D10-100
LT.1	K17	1	0,9	1,5	24D22	7D10-75	8D10-75
LT.1	K18-1	0,7	0,65	1,5	10D25	7D10-100	7D10-100
LT.1	K19-1	0,6	0,6	1,5	12D22	5D10-75	5D10-75
LT.1	K20	1,2	0,5	1,5	30D22	7D10-100	13D10-100
LT.1	K21	0,6	0,55	1,5	12D22	6D10-100	7D10-100
LT.1	K22	0,6	0,6	1,5	16D22	5D10-75	5D10-75
LT.2	K23	1	0,9	4,5	34D25	7D10-75	8D10-75
LT.2	K24	0,9	1,05	4,5	34D22	8D10-75	7D10-75
LT.2	K25	1	1	4,5	32D22	6D13-100	6D13-100
LT.2	K26	1,05	0,9	4,5	26D22	7D10-75	8D10-75
LT.2	K27	1	0,9	4,5	24D22	7D10-75	8D10-75
LT.2	K28	1	1,05	4,5	22D25	8D10-75	8D10-75
LT.2	K29	0,9	1	4,5	24D22	8D10-75	7D10-75
LT.2	K30	1,2	0,5	4,5	18D22	5D10-100	13D10-100
LT.2	K31	0,85	0,9	4,5	16D25	9D10-100	9D10-100
LT.2	K32	1,05	1,05	4,5	32D22	8D10-75	8D10-75
LT.2	K33	1,15	0,5	4,5	16D22	5D10-100	12D10-100
LT.2	K34	1,2	0,55	4,5	14D25	4D10-75	10D10-75
LT.2	K35	1,2	0,55	4,5	28D22	6D10-100	13D10-100
LT.2	K36-1	0,55	0,7	4,5	12D22	4D10-50	3D10-50
LT.2	K37-1	0,5	0,6	4,5	10D22	4D10-50	3D10-50
LT.2	K38	0,95	0,9	4,5	24D22	9D10-100	10D10-100
LT.2	K39	0,95	1	4,5	20D25	6D13-100	6D13-100
LT.2	K40-1	0,55	0,65	4,5	16D22	7D10-100	6D10-100
LT.2	K41-1	0,75	0,5	4,5	14D22	4D10-75	7D10-75
LT.2	K42	1,2	0,5	4,5	22D22	5D10-100	13D10-100
LT.2	K43	0,55	0,6	4,5	10D22	7D10-100	6D10-100
LT.2	K44	0,7	0,6	4,5	12D22	6D10-100	7D10-100
LT.3	K45	1,1	0,95	3,96	22D25	6D13-100	7D13-100
LT.3	K46	0,9	0,8	3,96	30D22	8D10-100	9D10-100
LT.3	K47	0,8	0,8	3,96	28D22	6D10-75	6D10-75

Tabel Hasil optimasi SOS *assisted* elemen struktur kolom studi kasus (Lanjutan)

Lantai	Nama ID Elemen	Panjang Kolom (m)	Lebar Kolom (m)	Tinggi Kolom(m)	Tulangan Longitudinal	Tulangan Sengkang arah y	Tulangan Sengkang arah x
LT.3	K48	0,85	1	3,96	24D22	6D13-100	5D13-100
LT.3	K49	0,85	0,85	3,96	24D22	5D13-100	5D13-100
LT.3	K50	0,9	0,95	3,96	24D22	10D10-100	9D10-100
LT.3	K51	1	0,85	3,96	24D22	5D13-100	6D13-100
LT.3	K52	1,2	0,5	3,96	22D22	5D10-100	13D10-100
LT.3	K53	0,85	0,9	3,96	24D22	9D10-100	9D10-100
LT.3	K54	0,85	0,85	3,96	20D22	5D13-100	5D13-100
LT.3	K55	1,2	0,5	3,96	18D22	5D10-100	13D10-100
LT.3	K56	1,15	0,5	3,96	20D22	5D10-100	12D10-100
LT.3	K57	1,2	0,5	3,96	18D22	5D10-100	13D10-100
LT.3	K58-1	0,5	0,7	3,96	12D22	6D10-75	4D10-75
LT.3	K59-1	0,75	0,6	3,96	12D22	3D10-50	4D10-50
LT.3	K60	0,9	0,8	3,96	20D22	8D10-100	9D10-100
LT.3	K61	0,95	0,95	3,96	24D22	5D10-50	5D10-50
LT.3	K62-1	0,6	0,65	3,96	18D22	7D10-100	6D10-100
LT.3	K63-1	0,75	0,6	3,96	20D22	6D10-100	8D10-100
LT.3	K64	1,2	0,5	3,96	22D22	5D10-100	13D10-100
LT.3	K65	0,85	0,7	0,6	18D22	7D10-100	9D10-100
LT.3	K66	0,6	0,6	3,36	16D22	5D10-75	5D10-75
LT.3	K67	0,65	0,6	0,6	8D25	6D10-100	7D10-100
LT.3	K68	0,6	0,6	3,36	8D25	5D10-75	5D10-75
LT.4	K69	0,85	0,9	3,96	24D22	9D10-100	9D10-100
LT.4	K70	0,95	0,75	3,96	32D22	8D10-100	10D10-100
LT.4	K71	0,75	0,95	3,96	32D22	10D10-100	8D10-100
LT.4	K72	0,9	0,8	3,96	22D22	8D10-100	9D10-100
LT.4	K73	0,75	0,8	3,96	24D22	6D10-75	6D10-75
LT.4	K74	1	0,75	3,96	20D22	5D13-100	6D13-100
LT.4	K75	0,85	0,8	3,96	20D22	5D13-100	5D13-100
LT.4	K76	1,15	0,6	3,96	20D22	6D10-100	12D10-100
LT.4	K77	0,8	0,85	3,96	24D22	5D13-100	5D13-100
LT.4	K78	1	0,85	3,96	24D22	5D13-100	6D13-100
LT.4	K79	1,2	0,5	3,96	18D22	5D10-100	13D10-100
LT.4	K80	1,2	0,55	3,96	20D22	6D10-100	13D10-100

Tabel Hasil optimasi SOS *assisted* elemen struktur kolom studi kasus (Lanjutan)

Lantai	Nama ID Elemen	Panjang Kolom (m)	Lebar Kolom (m)	Tinggi Kolom(m)	Tulangan Longitudinal	Tulangan Sengkang arah y	Tulangan Sengkang arah x
LT.4	K81	1,2	0,5	3,96	18D22	5D10-100	13D10-100
LT.4	K82-1	0,55	0,6	3,96	10D25	7D10-100	6D10-100
LT.4	K83-1	0,6	0,6	3,96	8D25	5D10-75	5D10-75
LT.4	K84	0,8	0,85	3,96	18D22	5D13-100	5D13-100
LT.4	K85	0,8	0,75	3,96	18D22	6D10-75	6D10-75
LT.4	K86-1	0,55	0,8	3,96	20D22	6D10-75	4D10-75
LT.4	K87-1	0,6	0,6	3,96	16D22	5D10-75	5D10-75
LT.4	K88	1,1	0,5	3,96	18D22	5D10-100	12D10-100
LT.5	K89	0,7	0,7	3,96	20D22	7D10-100	7D10-100
LT.5	K90	0,7	0,8	3,96	22D22	8D10-100	7D10-100
LT.5	K91	0,6	0,75	3,96	30D22	8D10-100	6D10-100
LT.5	K92	0,75	0,75	3,96	20D22	6D10-75	6D10-75
LT.5	K93	0,75	0,9	3,96	22D22	9D10-100	8D10-100
LT.5	K94	0,7	0,8	3,96	16D22	8D10-100	7D10-100
LT.5	K95	0,75	0,7	3,96	18D22	7D10-100	8D10-100
LT.5	K96	0,9	0,5	3,96	12D22	4D10-75	7D10-75
LT.5	K97	0,8	0,8	3,96	20D22	6D10-75	6D10-75
LT.5	K98	0,7	0,7	3,96	20D22	7D10-100	7D10-100
LT.5	K99	1	0,5	3,96	14D22	4D10-75	8D10-75
LT.5	K100	0,9	0,5	3,96	16D22	4D10-75	7D10-75
LT.5	K101	0,95	0,5	3,96	14D22	5D10-100	10D10-100
LT.5	K102-1	0,6	0,7	3,96	12D22	7D10-100	6D10-100
LT.5	K103-1	0,6	0,75	3,96	12D22	4D10-50	3D10-50
LT.5	K104	0,7	0,7	3,96	16D22	7D10-100	7D10-100
LT.5	K105	0,6	0,75	3,96	20D22	8D10-100	6D10-100
LT.5	K106-1	0,6	0,85	3,96	18D22	9D10-100	6D10-100
LT.5	K107-1	0,55	0,6	3,96	10D25	7D10-100	6D10-100
LT.5	K108	1,2	0,5	3,96	18D22	5D10-100	13D10-100
LT.6	K109	0,7	1	3,96	20D22	11D10-100	7D10-100
LT.6	K110	0,7	0,65	3,96	18D22	7D10-100	7D10-100
LT.6	K111	0,75	0,7	3,96	26D22	7D10-100	8D10-100

Tabel Hasil optimasi SOS *assisted* elemen struktur kolom studi kasus (Lanjutan)

Lantai	Nama ID Elemen	Panjang Kolom (m)	Lebar Kolom (m)	Tinggi Kolom(m)	Tulangan Longitudinal	Tulangan Sengkang arah y	Tulangan Sengkang arah x
LT.6	K112	0,75	0,75	3,96	20D22	6D10-75	6D10-75
LT.6	K113	0,8	0,8	3,96	28D22	6D10-75	6D10-75
LT.6	K114	0,7	0,85	3,96	18D22	9D10-100	7D10-100
LT.6	K115	0,7	0,7	3,96	24D22	7D10-100	7D10-100
LT.6	K116	0,9	0,5	3,96	12D22	4D10-75	7D10-75
LT.6	K117	0,7	0,85	3,96	18D22	9D10-100	7D10-100
LT.6	K118	0,85	0,75	3,96	22D25	5D13-100	5D13-100
LT.6	K119	1,15	0,5	3,96	16D22	5D10-100	12D10-100
LT.6	K120	0,9	0,5	3,96	12D22	4D10-75	7D10-75
LT.6	K121	1,1	0,55	3,96	18D22	4D10-75	9D10-75
LT.6	K122-1	0,65	0,7	3,96	12D22	7D10-100	7D10-100
LT.6	K123-1	0,65	0,55	3,96	10D22	6D10-100	7D10-100
LT.6	K124	0,9	0,7	3,96	18D22	7D10-100	9D10-100
LT.6	K125	0,85	0,8	3,96	18D22	5D13-100	5D13-100
LT.6	K126-1	0,65	0,65	3,96	12D22	5D10-75	5D10-75
LT.6	K127-1	0,55	0,6	3,96	10D22	7D10-100	6D10-100
LT.6	K128	0,95	0,5	3,96	14D22	5D10-100	10D10-100
DAK ATAP	K129	0,6	0,6	5,05	8D25	5D10-75	5D10-75
DAK ATAP	K130	0,75	0,55	5,05	14D22	4D10-75	6D10-75
DAK ATAP	K131	0,6	0,7	5,05	14D22	7D10-100	6D10-100
DAK ATAP	K132	0,65	0,55	5,05	14D22	6D10-100	7D10-100
DAK ATAP	K133	0,6	0,6	5,05	8D25	5D10-75	5D10-75
DAK ATAP	K134	0,6	0,6	5,05	8D25	5D10-75	5D10-75
DAK ATAP	K135	1	0,5	5,05	14D22	4D10-75	8D10-75

Tabel Hasil optimasi SOS *assisted* elemen struktur kolom studi kasus (Lanjutan)

Lantai	Nama ID Elemen	Panjang Kolom (m)	Lebar Kolom (m)	Tinggi Kolom(m)	Tulangan Longitudinal	Tulangan Sengkang arah y	Tulangan Sengkang arah x
DAK ATAP	K136	0,6	0,6	5,05	20D22	5D10-75	5D10-75
DAK ATAP	K137	1,05	0,5	5,05	16D22	4D10-75	8D10-75
DAK ATAP	K138	1,2	0,5	5,05	18D22	5D10-100	13D10-100
DAK ATAP	K139	1,05	0,5	5,05	16D22	4D10-75	8D10-75
DAK ATAP	K140-1	0,65	0,6	5,05	12D22	6D10-100	7D10-100
DAK ATAP	K141-1	0,65	0,6	5,05	8D25	6D10-100	7D10-100
DAK ATAP	K142	0,55	0,55	5,05	8D22	5D10-75	5D10-75
DAK ATAP	K143	0,55	0,6	5,05	10D22	7D10-100	6D10-100
DAK ATAP	K144-1	0,55	0,7	5,05	12D22	4D10-50	3D10-50
DAK ATAP	K145-1	0,55	0,6	5,05	10D22	7D10-100	6D10-100
DAK ATAP	K146	0,95	0,5	5,05	14D22	5D10-100	10D10-100

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.1	G1	0,35	0,7	4	6D19	3D19	5D19	6D19	4D19	5D19	2D10-75	2D10-75	2D10-75
LT.1	G2	0,25	0,75	11,0494	3D22	2D22	4D22	2D22	2D22	2D22	2D10-125	2D10-325	2D10-125
LT.1	G3	0,35	0,95	10,3	9D19	5D19	9D19	5D19	5D19	5D19	2D10-100	2D10-125	2D10-100
LT.1	G4	0,3	0,65	5,7	3D22	2D22	4D22	2D22	2D22	3D22	2D10-100	2D10-125	2D10-100
LT.1	G5	0,25	0,5	5,7	3D19	2D19	3D19	2D19	3D19	2D19	2D10-100	2D10-200	2D10-100
LT.1	G6	0,35	0,95	10,3	10D19	5D19	10D19	5D19	8D19	5D19	2D10-100	2D10-125	2D10-100
LT.1	G7	0,35	0,7	6,3	6D19	3D19	6D19	3D19	5D19	3D19	2D10-100	2D10-150	2D10-100
LT.1	G8	0,3	0,65	4	4D19	3D19	4D19	4D19	3D19	3D19	2D10-100	2D10-125	2D10-100
LT.1	G9	0,25	0,55	7,8	5D19	2D19	5D19	3D19	5D19	3D19	2D10-100	2D10-175	2D10-100
LT.1	G10	0,3	0,65	7,8	4D22	2D22	4D22	2D22	3D22	2D22	2D10-125	2D10-200	2D10-125
LT.1	G11	0,3	0,65	7,8	4D22	2D22	4D22	2D22	4D22	2D22	2D10-125	2D10-275	2D10-125
LT.1	G12	0,35	0,7	4,5	5D19	3D19	7D19	4D19	4D19	4D19	2D10-75	2D10-100	2D10-50
LT.1	G13	0,3	0,65	4,5	3D19	3D19	4D19	3D19	3D19	3D19	2D10-100	2D10-125	2D10-100
LT.1	G14	0,35	0,9	10,3	10D19	4D19	10D19	5D19	9D19	5D19	2D10-100	2D10-100	2D10-100
LT.1	G15	0,25	0,5	6,3	4D19	2D19	4D19	2D19	5D19	2D19	2D10-100	2D10-200	2D10-100
LT.1	G16	0,25	0,6	5,775	3D22	2D22	3D22	2D22	2D22	2D22	2D10-125	2D10-175	2D10-125

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.1	G17	0,3	0,65	4	3D22	2D22	3D22	3D22	2D22	3D22	2D10-125	2D10-125	2D10-125
LT.1	G18	0,25	0,5	3,1	2D19	2D19	2D19	2D19	2D19	2D19	2D10-100	2D10-125	2D10-100
LT.1	G19	0,25	0,5	3,1	3D19	2D19	2D19	2D19	2D19	2D19	2D10-100	2D10-100	2D10-100
LT.1	G20	0,25	0,5	2,5	3D19	2D19	3D19	3D19	2D19	2D19	2D10-100	2D10-150	2D10-100
LT.1	G21	0,3	0,65	6,4	3D22	2D22	3D22	2D22	2D22	2D22	2D10-125	2D10-175	2D10-125
LT.1	G22	0,25	0,55	6,4	3D19	2D19	4D19	2D19	3D19	2D19	2D10-100	2D10-200	2D10-100
LT.1	G23	0,25	0,5	6,4	3D19	2D19	3D19	2D19	2D19	2D19	2D10-100	2D10-200	2D10-100
LT.1	G24	0,25	0,5	3,3	2D19	2D19	3D19	2D19	2D19	2D19	2D10-100	2D10-150	2D10-100
LT.1	G25	0,25	0,5	3,3	3D19	2D19	3D19	3D19	2D19	2D19	2D10-100	2D10-150	2D10-100
LT.1	G26	0,25	0,75	10,3	6D19	3D19	6D19	3D19	5D19	3D19	2D10-100	2D10-200	2D10-100
LT.1	G27	0,25	0,5	2,5	3D19	2D19	3D19	3D19	2D19	3D19	2D10-100	2D10-150	2D10-100
LT.1	G28	0,25	0,5	3,275	3D19	2D19	3D19	3D19	2D19	2D19	2D10-100	2D10-150	2D10-100
LT.2	G29	0,5	1	4	14D19	9D19	18D19	14D19	9D19	18D19	3D10-50	3D10-50	3D10-50
LT.2	G30	0,45	1	11,0494	11D19	6D19	16D19	8D19	6D19	8D19	3D10-100	3D10-125	3D10-100
LT.2	G31	0,45	1	10,3	13D19	6D19	16D19	10D19	6D19	10D19	3D10-100	3D10-125	3D10-100
LT.2	G32	0,5	1	5,7	17D19	8D19	16D19	15D19	8D19	14D19	3D10-50	3D10-75	3D10-50
LT.2	G33	0,4	0,8	5,7	9D19	4D19	10D19	8D19	4D19	8D19	2D10-75	2D10-75	2D10-50

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.2	G34	0,45	1	10,3	15D19	6D19	18D19	8D19	7D19	9D19	3D10-100	3D10-125	3D10-100
LT.2	G35	0,5	1	6,3	15D19	7D19	17D19	15D19	8D19	12D19	3D10-75	3D10-75	3D10-50
LT.2	G36	0,4	0,8	4	9D19	4D19	10D19	9D19	4D19	9D19	2D10-50	2D10-50	2D10-50
LT.2	G37	0,5	1	7,8	11D22	5D22	12D22	8D22	5D22	8D22	3D10-75	3D10-100	3D10-75
LT.2	G38	0,35	0,8	7,8	9D19	4D19	9D19	5D19	4D19	5D19	2D10-75	2D10-100	2D10-75
LT.2	G39	0,4	0,8	7,8	10D19	4D19	12D19	6D19	7D19	6D19	2D10-50	2D10-75	2D10-75
LT.2	G40	0,45	0,9	4,5	10D19	5D19	15D19	11D19	6D19	10D19	3D10-75	3D10-50	3D10-50
LT.2	G41	0,4	0,8	4,5	7D22	4D22	8D22	7D22	4D22	6D22	2D10-50	2D10-25	2D10-25
LT.2	G42	0,45	1	10,3	17D19	6D19	17D19	9D19	8D19	9D19	3D10-100	3D10-125	3D10-100
LT.2	G43	0,4	0,8	6,3	9D19	4D19	12D19	7D19	5D19	7D19	2D10-50	2D10-75	2D10-50
LT.2	G44	0,3	0,65	5,775	4D22	2D22	4D22	3D22	2D22	3D22	2D10-100	2D10-125	2D10-100
LT.2	G45	0,35	0,7	4	9D19	3D19	6D19	7D19	4D19	6D19	2D10-50	2D10-75	2D10-50
LT.2	G46	0,3	0,65	3,1	4D22	2D22	4D22	4D22	2D22	4D22	2D10-100	2D10-100	2D10-100
LT.2	G47	0,25	0,5	3,1	4D19	2D19	2D19	2D19	2D19	2D19	2D10-75	2D10-100	2D10-100
LT.2	G48	0,35	0,7	2,5	7D19	3D19	7D19	8D19	3D19	6D19	2D10-50	2D10-50	2D10-50
LT.2	G49	0,5	1	6,4	17D19	7D19	15D19	14D19	8D19	13D19	3D10-75	3D10-75	3D10-75
LT.2	G50	0,35	0,7	6,4	7D19	3D19	7D19	5D19	4D19	5D19	2D10-100	2D10-100	2D10-100

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.2	G51	0,3	0,7	6,4	5D22	2D22	6D22	4D22	2D22	4D22	2D10-125	2D10-125	2D10-125
LT.2	G52	0,35	0,7	6,4	6D19	3D19	7D19	5D19	3D19	4D19	2D10-100	2D10-125	2D10-75
LT.2	G53	0,25	0,5	3,3	4D19	2D19	3D19	3D19	2D19	3D19	2D10-100	2D10-125	2D10-100
LT.2	G54	0,3	0,65	3,3	4D22	2D22	4D22	4D22	2D22	4D22	2D10-75	2D10-75	2D10-75
LT.2	G55	0,45	0,9	10,3	10D22	4D22	10D22	5D22	4D22	5D22	3D10-100	3D10-125	3D10-100
LT.2	G56	0,35	0,7	6,3	6D19	3D19	8D19	5D19	3D19	5D19	2D10-100	2D10-100	2D10-75
LT.2	G57	0,3	0,65	2,5	6D19	3D19	6D19	5D19	3D19	6D19	2D10-75	2D10-75	2D10-75
LT.2	G58	0,35	0,7	3,275	8D19	3D19	6D19	7D19	3D19	6D19	2D10-100	2D10-100	2D10-100
LT.3	G59	0,5	1	4	14D19	8D19	18D19	14D19	8D19	17D19	3D10-50	3D10-50	3D10-50
LT.3	G60	0,45	1	11,0494	11D19	6D19	16D19	8D19	6D19	8D19	3D10-100	3D10-125	3D10-100
LT.3	G61	0,45	1	10,3	13D19	6D19	16D19	10D19	6D19	9D19	3D10-100	3D10-125	3D10-100
LT.3	G62	0,45	0,9	5,7	14D19	5D19	13D19	12D19	6D19	11D19	3D10-75	3D10-75	3D10-75
LT.3	G63	0,35	0,7	5,7	7D19	3D19	8D19	6D19	4D19	6D19	2D10-75	2D10-100	2D10-75
LT.3	G64	0,5	1	10,3	16D19	7D19	18D19	8D19	7D19	9D19	3D10-100	3D10-100	3D10-100
LT.3	G65	0,45	0,9	6,3	10D22	4D22	11D22	9D22	4D22	7D22	3D10-75	3D10-75	3D10-75
LT.3	G66	0,45	0,9	4	11D19	5D19	13D19	12D19	6D19	11D19	3D10-100	3D10-75	3D10-75
LT.3	G67	0,5	1	7,8	12D22	5D22	13D22	9D22	6D22	9D22	3D10-75	3D10-75	3D10-75

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.3	G68	0,35	0,7	7,8	8D19	3D19	8D19	5D19	4D19	4D19	2D10-75	2D10-100	2D10-100
LT.3	G69	0,45	0,9	7,8	11D19	5D19	14D19	8D19	7D19	7D19	3D10-75	3D10-100	3D10-100
LT.3	G70	0,5	1	4,5	13D19	7D19	17D19	15D19	8D19	12D19	3D10-75	3D10-50	3D10-50
LT.3	G71	0,45	0,9	4,5	8D22	4D22	9D22	8D22	4D22	7D22	3D10-100	3D10-75	3D10-75
LT.3	G72	0,45	1	10,3	17D19	6D19	16D19	9D19	7D19	8D19	3D10-100	3D10-125	3D10-100
LT.3	G73	0,45	0,9	6,3	10D19	5D19	13D19	8D19	5D19	9D19	3D10-100	3D10-100	3D10-75
LT.3	G74	0,3	0,65	5,775	6D19	3D19	5D19	4D19	3D19	3D19	2D10-100	2D10-125	2D10-100
LT.3	G75	0,35	0,7	4	8D19	3D19	7D19	6D19	4D19	7D19	2D10-50	2D10-75	2D10-50
LT.3	G76	0,3	0,65	3,1	5D19	3D19	5D19	5D19	3D19	5D19	2D10-100	2D10-100	2D10-100
LT.3	G77	0,3	0,65	3,1	5D19	3D19	3D19	3D19	3D19	3D19	2D10-75	2D10-75	2D10-100
LT.3	G78	0,3	0,65	2,5	4D22	2D22	4D22	4D22	2D22	4D22	2D10-75	2D10-50	2D10-50
LT.3	G79	0,5	1	6,4	18D19	8D19	15D19	15D19	8D19	14D19	3D10-50	3D10-75	3D10-75
LT.3	G80	0,35	0,7	6,4	4D25	2D25	5D25	3D25	2D25	3D25	2D10-100	2D10-100	2D10-100
LT.3	G81	0,4	0,85	6,4	6D25	3D25	6D25	6D25	3D25	5D25	2D10-75	2D10-75	2D10-75
LT.3	G82	0,4	0,8	6,4	9D19	5D19	11D19	9D19	6D19	8D19	2D10-75	2D10-75	2D10-50
LT.3	G83	0,25	0,75	6,4	2D22	2D22	2D22	2D22	2D22	2D22	2D10-125	2D10-300	2D10-125
LT.3	G84	0,3	0,65	3,3	5D19	3D19	5D19	6D19	3D19	4D19	2D10-75	2D10-75	2D10-50

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.3	G85	0,3	0,65	3,3	5D19	3D19	6D19	6D19	3D19	5D19	2D10-50	2D10-50	2D10-50
LT.3	G86	0,45	0,9	10,3	8D25	3D25	7D25	4D25	3D25	4D25	3D10-100	3D10-125	3D10-125
LT.3	G87	0,35	0,7	6,3	4D25	2D25	5D25	3D25	2D25	3D25	2D10-100	2D10-100	2D10-75
LT.3	G88	0,35	0,7	2,5	6D22	4D22	5D22	5D22	3D22	5D22	2D10-50	2D10-50	2D10-50
LT.3	G89	0,35	0,7	3,275	8D19	3D19	6D19	7D19	3D19	6D19	2D10-75	2D10-75	2D10-100
LT.3	G90	0,35	0,7	4,7	7D19	4D19	7D19	6D19	4D19	6D19	2D10-75	2D10-75	2D10-75
LT.3	G91	0,4	0,8	4,7	8D19	5D19	11D19	8D19	5D19	10D19	2D10-50	2D10-75	2D10-50
LT.3	G92	0,3	0,65	2,6	6D19	3D19	5D19	5D19	3D19	5D19	2D10-75	2D10-75	2D10-75
LT.3	G93	0,25	0,5	2,6	4D19	2D19	3D19	3D19	2D19	3D19	2D10-100	2D10-125	2D10-100
LT.4	G94	0,45	0,9	4	14D19	6D19	15D19	14D19	6D19	14D19	3D10-75	3D10-75	3D10-75
LT.4	G95	0,45	1	11,0494	10D19	6D19	15D19	7D19	6D19	8D19	3D10-100	3D10-150	3D10-100
LT.4	G96	0,45	1	10,3	12D19	6D19	16D19	9D19	6D19	8D19	3D10-100	3D10-125	3D10-100
LT.4	G97	0,4	0,8	5,7	12D19	5D19	13D19	10D19	6D19	10D19	2D10-50	2D10-50	2D10-50
LT.4	G98	0,35	0,7	5,7	7D19	3D19	8D19	6D19	4D19	6D19	2D10-75	2D10-100	2D10-75
LT.4	G99	0,45	1	10,3	15D19	6D19	18D19	8D19	7D19	9D19	3D10-100	3D10-125	3D10-100
LT.4	G100	0,45	0,9	6,3	9D22	4D22	11D22	8D22	4D22	7D22	3D10-75	3D10-75	3D10-75
LT.4	G101	0,4	0,8	4	7D22	4D22	7D22	7D22	4D22	6D22	2D10-50	2D10-50	2D10-50

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.4	G102	0,45	0,9	7,8	15D19	5D19	15D19	10D19	6D19	10D19	3D10-75	3D10-100	3D10-75
LT.4	G103	0,4	0,8	7,8	10D19	4D19	12D19	6D19	5D19	6D19	2D10-50	2D10-75	2D10-50
LT.4	G104	0,45	0,9	7,8	8D22	4D22	11D22	6D22	5D22	6D22	3D10-75	3D10-100	3D10-75
LT.4	G105	0,5	1	4,5	12D19	7D19	16D19	13D19	7D19	10D19	3D10-75	3D10-50	3D10-50
LT.4	G106	0,4	0,8	4,5	6D22	3D22	8D22	7D22	4D22	6D22	2D10-50	2D10-25	2D10-25
LT.4	G107	0,45	1	10,3	17D19	6D19	17D19	9D19	8D19	9D19	3D10-100	3D10-100	3D10-100
LT.4	G108	0,4	0,8	6,3	9D19	4D19	11D19	6D19	4D19	6D19	2D10-75	2D10-75	2D10-50
LT.4	G109	0,35	0,7	5,775	6D19	3D19	6D19	5D19	3D19	4D19	2D10-100	2D10-100	2D10-100
LT.4	G110	0,35	0,7	4	8D19	3D19	6D19	6D19	3D19	6D19	2D10-50	2D10-75	2D10-75
LT.4	G111	0,3	0,65	3,1	4D22	2D22	4D22	4D22	2D22	4D22	2D10-100	2D10-100	2D10-100
LT.4	G112	0,3	0,65	3,1	4D22	3D22	2D22	3D22	3D22	2D22	2D10-75	2D10-75	2D10-75
LT.4	G113	0,3	0,65	2,5	5D19	3D19	5D19	5D19	3D19	4D19	2D10-75	2D10-75	2D10-75
LT.4	G114	0,45	0,9	6,4	12D22	5D22	10D22	9D22	5D22	9D22	3D10-75	3D10-75	3D10-75
LT.4	G115	0,35	0,7	6,4	4D25	2D25	5D25	3D25	2D25	3D25	2D10-75	2D10-100	2D10-75
LT.4	G116	0,35	0,7	6,4	7D19	3D19	8D19	6D19	4D19	5D19	2D10-100	2D10-100	2D10-75
LT.4	G117	0,35	0,7	3,3	6D19	3D19	6D19	7D19	3D19	5D19	2D10-100	2D10-75	2D10-75
LT.4	G118	0,3	0,65	3,3	5D19	3D19	6D19	6D19	3D19	5D19	2D10-75	2D10-75	2D10-50

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.4	G119	0,4	1	10,3	12D19	5D19	12D19	6D19	5D19	6D19	2D10-100	2D10-125	2D10-100
LT.4	G120	0,35	0,75	6,3	4D25	2D25	5D25	3D25	2D25	3D25	2D10-75	2D10-100	2D10-75
LT.4	G121	0,4	0,8	2,5	6D22	3D22	5D22	5D22	3D22	5D22	2D10-50	2D10-50	2D10-50
LT.4	G122	0,3	0,65	3,275	4D22	2D22	3D22	4D22	2D22	3D22	2D10-75	2D10-100	2D10-75
LT.5	G123	0,4	0,8	4	5D25	3D25	7D25	5D25	3D25	6D25	2D10-50	2D10-25	2D10-25
LT.5	G124	0,4	1	11,0494	8D19	5D19	13D19	5D19	5D19	7D19	2D10-100	2D10-125	2D10-100
LT.5	G125	0,4	1	10,3	10D19	5D19	14D19	6D19	5D19	7D19	2D10-100	2D10-100	2D10-100
LT.5	G126	0,4	0,8	5,7	12D19	4D19	12D19	10D19	5D19	9D19	2D10-50	2D10-50	2D10-50
LT.5	G127	0,3	0,65	5,7	4D22	2D22	5D22	3D22	2D22	3D22	2D10-100	2D10-100	2D10-100
LT.5	G128	0,45	1	10,3	15D19	6D19	17D19	8D19	7D19	9D19	3D10-100	3D10-125	3D10-100
LT.5	G129	0,4	0,8	6,3	9D19	4D19	12D19	7D19	5D19	6D19	2D10-75	2D10-50	2D10-50
LT.5	G130	0,35	0,7	4	6D19	3D19	6D19	6D19	3D19	5D19	2D10-75	2D10-75	2D10-75
LT.5	G131	0,4	0,8	7,8	13D19	4D19	13D19	8D19	5D19	7D19	2D10-50	2D10-75	2D10-50
LT.5	G132	0,4	0,8	7,8	9D19	4D19	11D19	6D19	4D19	6D19	2D10-75	2D10-75	2D10-75
LT.5	G133	0,45	0,9	7,8	10D19	5D19	15D19	8D19	7D19	8D19	3D10-75	3D10-100	3D10-75
LT.5	G134	0,45	0,9	4,5	8D19	5D19	13D19	9D19	5D19	7D19	3D10-75	3D10-50	3D10-50
LT.5	G135	0,4	0,8	4,5	8D19	4D19	10D19	9D19	4D19	6D19	2D10-50	2D10-50	2D10-50

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.5	G136	0,45	1	10,3	15D19	6D19	15D19	8D19	8D19	8D19	3D10-100	3D10-125	3D10-100
LT.5	G137	0,4	0,8	6,3	7D19	4D19	10D19	5D19	4D19	5D19	2D10-75	2D10-75	2D10-75
LT.5	G138	0,3	0,65	5,775	5D19	3D19	5D19	3D19	3D19	3D19	2D10-100	2D10-125	2D10-100
LT.5	G139	0,3	0,65	4	4D22	2D22	3D22	3D22	2D22	3D22	2D10-100	2D10-100	2D10-100
LT.5	G140	0,3	0,65	3,1	4D22	2D22	4D22	4D22	2D22	4D22	2D10-100	2D10-100	2D10-100
LT.5	G141	0,3	0,65	3,1	4D22	2D22	2D22	3D22	2D22	2D22	2D10-75	2D10-75	2D10-100
LT.5	G142	0,25	0,5	2,5	3D19	2D19	3D19	3D19	2D19	3D19	2D10-100	2D10-150	2D10-100
LT.5	G143	0,4	0,8	6,4	13D19	5D19	12D19	9D19	5D19	9D19	2D10-50	2D10-50	2D10-50
LT.5	G144	0,35	0,7	6,4	7D19	3D19	8D19	5D19	4D19	4D19	2D10-75	2D10-100	2D10-75
LT.5	G145	0,35	0,7	6,4	6D19	3D19	8D19	6D19	4D19	4D19	2D10-100	2D10-100	2D10-75
LT.5	G146	0,3	0,65	3,3	4D22	2D22	4D22	4D22	2D22	3D22	2D10-125	2D10-100	2D10-100
LT.5	G147	0,3	0,65	3,3	4D19	3D19	5D19	5D19	3D19	4D19	2D10-75	2D10-75	2D10-75
LT.5	G148	0,35	0,9	10,3	11D19	4D19	11D19	6D19	5D19	6D19	2D10-100	2D10-100	2D10-100
LT.5	G149	0,35	0,75	6,3	6D22	3D22	7D22	4D22	3D22	4D22	2D10-75	2D10-100	2D10-75
LT.5	G150	0,3	0,65	2,5	5D19	3D19	5D19	5D19	3D19	5D19	2D10-75	2D10-75	2D10-75
LT.5	G151	0,3	0,65	3,275	5D19	3D19	4D19	5D19	3D19	4D19	2D10-75	2D10-100	2D10-75
LT.6	G152	0,3	0,65	4	5D19	3D19	7D19	5D19	3D19	5D19	2D10-75	2D10-75	2D10-75

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.6	G153	0,35	1	11,0494	6D19	5D19	11D19	5D19	5D19	6D19	2D10-100	2D10-150	2D10-100
LT.6	G154	0,35	1	10,3	8D19	5D19	12D19	6D19	5D19	6D19	2D10-100	2D10-100	2D10-75
LT.6	G155	0,35	0,7	5,7	9D19	3D19	9D19	6D19	4D19	6D19	2D10-75	2D10-75	2D10-75
LT.6	G156	0,3	0,65	5,7	5D19	3D19	6D19	4D19	3D19	4D19	2D10-100	2D10-125	2D10-100
LT.6	G157	0,35	0,95	10,3	10D19	5D19	11D19	5D19	6D19	6D19	2D10-100	2D10-125	2D10-100
LT.6	G158	0,35	0,7	6,3	6D19	3D19	8D19	4D19	3D19	4D19	2D10-100	2D10-100	2D10-75
LT.6	G159	0,3	0,65	4	4D19	3D19	5D19	4D19	3D19	3D19	2D10-100	2D10-100	2D10-100
LT.6	G160	0,35	0,85	7,8	11D19	4D19	11D19	6D19	5D19	6D19	2D10-75	2D10-100	2D10-100
LT.6	G161	0,3	0,75	7,8	6D19	3D19	7D19	4D19	3D19	4D19	2D10-100	2D10-150	2D10-100
LT.6	G162	0,4	0,8	7,8	6D22	3D22	8D22	4D22	4D22	4D22	2D10-75	2D10-75	2D10-75
LT.6	G163	0,4	0,8	4,5	4D22	3D22	8D22	5D22	3D22	4D22	2D10-50	2D10-50	2D10-25
LT.6	G164	0,35	0,7	4,5	5D19	3D19	7D19	5D19	3D19	4D19	2D10-75	2D10-75	2D10-50
LT.6	G165	0,35	0,95	10,3	8D22	4D22	8D22	4D22	6D22	4D22	2D10-100	2D10-100	2D10-100
LT.6	G166	0,35	0,7	6,3	6D19	3D19	8D19	3D19	4D19	4D19	2D10-100	2D10-125	2D10-100
LT.6	G167	0,3	0,65	5,775	5D19	3D19	6D19	3D19	3D19	3D19	2D10-100	2D10-100	2D10-100
LT.6	G168	0,3	0,65	4	3D25	2D25	2D25	2D25	2D25	2D25	2D10-100	2D10-125	2D10-100
LT.6	G169	0,25	0,5	3,1	4D19	2D19	3D19	3D19	2D19	3D19	2D10-75	2D10-75	2D10-75

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
LT.6	G170	0,3	0,65	3,1	4D22	2D22	2D22	2D22	2D22	2D22	2D10-75	2D10-75	2D10-75
LT.6	G171	0,25	0,5	2,5	2D19	2D19	3D19	3D19	2D19	2D19	2D10-100	2D10-125	2D10-100
LT.6	G172	0,35	0,7	6,4	9D19	3D19	8D19	6D19	4D19	5D19	2D10-75	2D10-75	2D10-75
LT.6	G173	0,3	0,7	6,4	3D25	2D25	4D25	2D25	2D25	2D25	2D10-100	2D10-125	2D10-100
LT.6	G174	0,35	0,7	6,4	6D19	3D19	8D19	6D19	4D19	4D19	2D10-100	2D10-100	2D10-75
LT.6	G175	0,3	0,65	3,3	4D19	3D19	5D19	5D19	3D19	4D19	2D10-75	2D10-75	2D10-75
LT.6	G176	0,25	0,5	3,3	3D19	2D19	4D19	3D19	2D19	2D19	2D10-100	2D10-100	2D10-100
LT.6	G177	0,35	1	10,3	9D19	5D19	9D19	5D19	5D19	5D19	2D10-100	2D10-150	2D10-100
LT.6	G178	0,3	0,65	6,3	4D22	2D22	4D22	3D22	2D22	2D22	2D10-100	2D10-125	2D10-100
LT.6	G179	0,25	0,5	2,5	3D19	2D19	3D19	3D19	2D19	3D19	2D10-100	2D10-150	2D10-100
LT.6	G180	0,25	0,5	3,275	3D19	2D19	3D19	3D19	2D19	3D19	2D10-100	2D10-150	2D10-100
DAK ATAP	G181	0,25	0,5	4	4D19	2D19	3D19	2D19	2D19	4D19	2D10-100	2D10-125	2D10-100
DAK ATAP	G182	0,3	0,9	11,0494	2D25	2D25	4D25	2D25	2D25	2D25	2D10-125	2D10-225	2D10-125
DAK ATAP	G183	0,35	1	10,3	5D19	5D19	10D19	5D19	5D19	5D19	2D10-100	2D10-150	2D10-100

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
DAK ATAP	G184	0,35	0,7	5,7	9D19	3D19	7D19	5D19	6D19	4D19	2D10-75	2D10-75	2D10-75
DAK ATAP	G185	0,25	0,5	4	2D19	2D19	3D19	2D19	2D19	2D19	2D10-100	2D10-200	2D10-100
DAK ATAP	G186	0,25	0,75	7,8	4D22	2D22	4D22	2D22	2D22	2D22	2D10-125	2D10-200	2D10-125
DAK ATAP	G187	0,35	0,7	7,8	4D19	3D19	9D19	4D19	3D19	5D19	2D10-100	2D10-125	2D10-100
DAK ATAP	G188	0,3	0,65	4,5	3D19	3D19	6D19	4D19	3D19	3D19	2D10-100	2D10-75	2D10-75
DAK ATAP	G189	0,3	0,65	4,5	2D22	2D22	4D22	3D22	2D22	2D22	2D10-100	2D10-100	2D10-75
DAK ATAP	G190	0,3	0,65	5,775	4D22	2D22	4D22	2D22	3D22	2D22	2D10-125	2D10-150	2D10-125
DAK ATAP	G191	0,25	0,55	4	3D19	2D19	3D19	2D19	2D19	2D19	2D10-100	2D10-175	2D10-100
DAK ATAP	G192	0,25	0,5	3,1	3D19	2D19	3D19	3D19	2D19	3D19	2D10-100	2D10-100	2D10-100
DAK ATAP	G193	0,25	0,5	3,1	4D19	2D19	2D19	2D19	2D19	2D19	2D10-100	2D10-100	2D10-100

Tabel Hasil optimasi SOS *assisted* elemen struktur balok studi kasus (Lanjutan)

Lantai	ID Elemen	Lebar Balok (m)	Tinggi Balok (m)	Bentang Balok (m)	Tulangan Tumpuan-I Atas	Tulangan Lapangan Atas	Tulangan Tumpuan-J Atas	Tulangan Tumpuan-I Bawah	Tulangan Lapangan Bawah	Tulangan Tumpuan-J Bawah	Sengkang Tumpuan-I	Sengkang Lapangan	Sengkang Tumpuan-J
DAK ATAP	G194	0,25	0,5	2,5	2D19	2D19	3D19	2D19	3D19	2D19	2D10-100	2D10-100	2D10-100
DAK ATAP	G195	0,3	0,7	6,4	4D22	2D22	3D22	2D22	2D22	3D22	2D10-125	2D10-150	2D10-125
DAK ATAP	G196	0,3	0,65	6,4	4D19	3D19	5D19	3D19	3D19	3D19	2D10-100	2D10-200	2D10-100
DAK ATAP	G197	0,25	0,5	3,3	3D19	2D19	3D19	3D19	3D19	2D19	2D10-100	2D10-100	2D10-75
DAK ATAP	G198	0,25	0,5	3,3	2D19	2D19	3D19	2D19	3D19	2D19	2D10-100	2D10-125	2D10-100
DAK ATAP	G199	0,25	0,75	10,3	3D22	2D22	5D22	2D22	2D22	3D22	2D10-125	2D10-200	2D10-125
DAK ATAP	G200	0,25	0,55	6,3	3D19	2D19	3D19	2D19	2D19	2D19	2D10-100	2D10-225	2D10-100
DAK ATAP	G201	0,25	0,5	2,5	3D19	2D19	3D19	2D19	3D19	2D19	2D10-100	2D10-150	2D10-100
DAK ATAP	G202	0,25	0,55	3,275	3D19	2D19	2D19	2D19	2D19	2D19	2D10-100	2D10-175	2D10-100

Lampiran 27: Contoh Perhitungan

Balok G2 (Momen Tumpuan Kanan)

b = 0.65 m

h = 1 m

Mu = 1082.8 kNm

d = h - 0.06 = 0.94 m

fc' = 40000 kN/m^2

fy = 420000 kN/m^2

Ln = 10.106 m

a = $(d - \sqrt{(d^2 - 2 * Mu/(0.85 * fc' * \theta * b)}$

a = $(0.94 - \sqrt{(0.94^2 - 2 * 1082.8/(0.85 * 40000 * 0.9 * 0.65)}$

a = 0.06 m

cmax = $\Sigma cmax/(\Sigma cmax + \Sigma smin)xd$

cmax = $0.003/(0.003 + 0.005)x0.94$

cmax = 0.3525 m

$\beta 1 = 0.85 - 0.05 * (fc - 28000)/7000$

$\beta 1$ = 0.7643

a max = $\beta 1\ x\ cmax$

a max = $0.7643\ x\ 0.3525$

untuk a < a max

$As = Mu/\left(\theta xfyx(d - a/2)\right)$

$As = 1082.8/\left(0.9x420000x(0.94 - 0.06/2)\right)$

$As\ =\ 0.0031\ m^2$

Ukuran tulangan terpasang:

Ukuran Tulangan (1) : 12D19 = 0.0034 m^2

Ukuran Tulangan (2) : 9D22 = 0.0034 m^2

Ukuran Tulangan (3) : 7D25 = 0.0034 m^2

Max Tulangan per Lapis (1) = ((0.65-0.008-0.0013x2+0.025)/(0.025+19/1000) = 12

Max Tulangan per Lapis (2) = ((0.65-0.008-0.0013x2+0.025)/(0.025+22/1000) = 12

Max Tulangan per Lapis (3) = ((0.65-0.008-0.0013x2+0.025)/(0.025+25/1000) = 11

Tulanga Lapis Akhir (1) = Mod(12,12) = 0, = 12

Tulangan Lapis Akhir (2) = Mod(9,12) = 9

Tulangan Lapis Akhir (3) = Mod(7,11) = 7

Lampiran 27: Contoh Perhitungan (Lanjutan)

Jumlah Lapis (1) = Ceil (12/12) = 1

Jumlah Lapis (2) = Ceil (9/12) = 1

Jumlah Lapis (3) = Ceil (7/11) = 1

dmax lapis (1) = 0.04+0.01+19/2000 = 0.0595 m

dmax lapis (2) = 0.04+0.01+22/2000 = 0.061 m

dmax lapis (3) = 0.04+0.01+22/2000 = 0.0625 m

Hitung As dengan d yang baru

Asmin1 = $0.25\ x\ \sqrt{fc'/1000}\ x\ b\ x\ d\ /\ fy\ x\ 1000$

Asmin2 = $1.4\ x\ b\ x\ d\ /\ fy\ x\ 1000$

Asmin = max(Asmin1,Asmin2)

Asmin(1) = 0.00315

Asmin(2) = 0.00315

Asmin(3) = 0.00316

Jumlah tulangan baru = Asmin/Asbar

Jumlah tulangan baru(1) = 0.00315/$(0.25x3.14x(19/1000)^2)$ = 12

Jumlah tulangan baru(2) = 0.00315/$(0.25x3.14x(22/1000)^2)$ = 9

Jumlah tulangan baru(3) = 0.00316/$(0.25x3.14x(25/1000)^2)$ = 7

As pasang = Asbar x Jumlah tulangan baru

As pasang(1) = 12 x $(0.25x3.14x(19/1000)^2)$ = 0.0034

As pasang(2) = 9 x $(0.25x3.14x(22/1000)^2)$ = 0.00342

As pasang(3) = 7 x $(0.25x3.14x(25/1000)^2)$ = 0.00344

Diambil As terkecil yaitu 12D19

Tulangan Geser

Ln = 10.106 m

Fy = 52500 kN/M^2

ϕ = 1

Lampiran 27: Contoh Perhitungan (Lanjutan)

	Tumpuan atas-I	Tumpuan bawah-I	Tumpuan atas-J	Tumpuan bawah - J
	9D19	9D19	12D19	9D19
As	0.0026	0.0026	0.0034	0.0026
d	0.9405	0.9405	0.9405	0.9405
a	0.0606	0.0606	0.0808	0.0606
Mpr	1219.3581	1219.3581	1607.7643	1219.3581

Contoh perhitungan

As = 9 x 0.25 x 3.14 x $(19/1000)^2$ = 0.0026 m^2

d = h-d1 = 1 - 0.0595

a = $AsxFy/(0.85xfc'xb)$

a = $0.0026x525000/(0.85xfc'xb)$ = 0.0606 m

Mpr = $Asxfyx(d-a/2) = 0.0026x52500x(0.0595-0.0606/2)$ = 1219.3581 kNm

Vp1 = Mpr Tumpuan atas-I + Mpr Tumpuan bawah-J /Ln

Vp1 = (1219.3581+1219.581)/10.106

Vp1 = 241.31 kN

Vp2 = Mpr Tumpuan bawah-I + Mpr Tumpuan atas-J /Ln

Vp2 = (1219.3581+1607.7643)/10.106

Vp2 = 279.75 kN

Vg = 265.61 kN

Vu = 385.603 kN

Ve = Vp2 + Vg = 279.75+265.61 = 545.36 kN

Karena Ve > ½ Vu, Vc diabaikan

Vc = 0 kN

Vs = Ve = 545.36 kN

Av/s = Vs/(0.75 x fy x d/1000)

Av/s = 545.36/(0.75 x 420000 x 0.94x1000)

Av/s = 1.84181 mm^2/mm

roundup(b-(dc x dsengkang) x 2 / smax sengkang

roundup(0.65 - (0.04x0.1)x2 / 0.35) = 3

Lampiran 27: Contoh Perhitungan (Lanjutan)

Perlu minimal 3 kaki

s = $nxdsengkangx3.14x0.25/(Av/s)x1000$

s = 3 x 10 x 3.14 x 0.25 / (1.84181)x1000

s = 0.127 mm

s pakai = 125 mm

Ash Kolom

fc' 40 MPa

fyt = 420 MPa

dt = 10 mm

h = 1150 mm

b = 950 mm

s = 50 mm

$Av \ = 0.25 \ x \ 3.14 \ x \ 10^2$

$Av \ = \ 78.539 \ mm^2$

$Ag \ = \ 1150x950 \ = \ 1092500 \ mm^2$

$Ach = (1150 - 2x40)x(950 - 2x40)$

$Ach \ = 930900 \ mm^2$

cek $Pu > 0.3xAgxfc'$

$6381000 \ > \ 0.3 \ x \ 1092500 \ x \ 40$

6381000 N > 13110000 N (*False*)

		Av/s		Legs	
		Depth	Width	Depth	Width
(a)	0.00496	5307.06	4315.09	3.37858	2.74707
(b)	0.00857	9171.43	7457.14	5.83871	4.74736
				5.83871	4.74736

Contoh perhitungan

$$Ash/sbc1 = 0.3\left(\frac{Ag}{Ach} - 1\right)\frac{fc'}{fyt}$$

$$Ash/sbc1 = 0.3\left(\frac{1092500}{930900} - 1\right)\frac{40}{420} = 0.00534$$

Lampiran 27: Contoh Perhitungan (Lanjutan)

$$Ash/sbc2 \; = \; 0.9\frac{fc'}{fyt}$$

$$Ash/sbc2 \; = \; 0.9\frac{40}{420} \; = 0.00857$$

$$Ash/sbc(max) = Ash/sbc2 = 0.9\frac{40}{420} = 0.00857$$

$$Av/s = 0.00857x(h - 2xdc)x1000$$

$$Av/s(1) = 0.00857x(1000 - 2x40)x1000 = 7885.71mm^2/mm$$

$$Av/s(2) = 0.00857x(950 - 2x40)x1000 = 7457.14mm^2/mm$$

$$Legs(1) = Av/s(1)xs/Av/1000$$

$$Legs(1) = 7885.71x50/78.53/1000 = 5.0202m$$

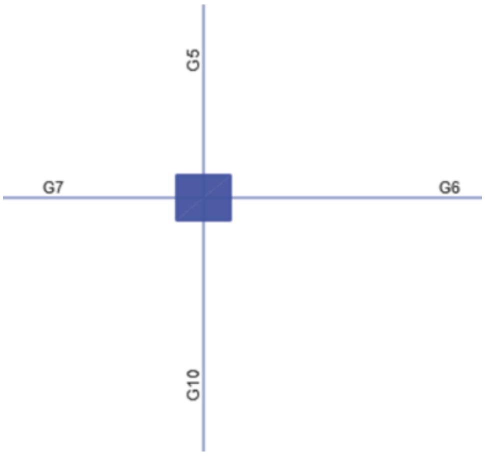
$$Legs(2) = Av/s(2)xs/Av/1000$$

$$Legs(2) = 7457.14x50/78.53/1000 = 4.74736m$$

Maj = roundup(Legs(1)) = 6D10-50

Min = roundup(Legs(2)) = 5D10-50

Contoh Perhitungan Joint Shear



Gambar kolom 5

Dimensi:

Panjang (h) = 1.15 m

Lebar (b) = 0.95 m

Luas = h x b = 1.15 x 0.95 = 1.092 m^2

Lampiran 27: Contoh Perhitungan (Lanjutan)

Balok yang merangkai pada kolom		
Kiri	G7	Mayor
Kanan	G6	Mayor
Atas	G5	Minor
Bawah	G10	Minor

						1.25 As Fy (kN)		a (m)		MPr (kNm)	
		Lebar	Tinggi	Tump. atas	Tump. bawah	Tump. atas	Tump. bawah	Atas	Bawah	Atas	Bawah
Kiri	G7	0.65	0.85	9D19	7D19	1339.6	1041,9	0.0606	0.0471	1018.4	799.11
Kanan	G6	0.65	1.05	7D25	5D25	1803.9	1288,5	0.0816	0.0583	1707.7	1234.8
Atas	G5	0.65	0.85	4D25	4D25	1030.8	1030,8	0.0466	0.0466	787.74	787.74
Bawah	G10	0.6	0.95	6D22	5D22	1197.4	997.84	0.0587	0.0489	969.49	812.79

Contoh perhitungan:

1.25 As Fy (kN)

Tump. atas = $0.25\ x\ 3.14\ x\ 19^2\ x\ 420\ /1000\ x\ 1.25\ x\ 9$

Tump atas = 1339.673 mm^2

Tump bawah = $0.25\ x\ 3.14\ x\ 19^2\ x\ 420\ /1000\ x\ 1.25$ x 7

Tump bawah = 1041.968 mm^2

a = Tumpuan atas/(0.85 x fc’ x b)

a (atas)= 1339.673/(0.85x40000x0.65) = 0.060619 m

a (bawah) = Tumpuan bawah/(0.85 x fc’ x b)

a (atas)= 1041.9681/(0.85x40000x0.65) = 0.0471 m

MPr = Tumpuan atas x (tinggi-d-a/2)

MPr = 1339.683 x (0.85-0.595-0.06062/2) = 1018,41 kNm

MPr = Tumpuan bawah x (tinggi-d-a/2)

MPr = 1041.968 x (0.85-0.565-0.0471/2) = 7911,112 kNm

Lampiran 27: Contoh Perhitungan (Lanjutan)

Arah major	Arah	C+T (kN)	Vu kolom(kN)	Vuj (kN)	Kapasitas Joint (kN)	D/C Ratio
G7 atas + G6 bawah	searah jarum jam	2628,217	751,0933	1877,123	5873,1401	0.319
G7 atas + G6 atas arah minor	berlawanan jarum jam	2845,93	835,6327	2010,296	5873,1401	0.342
Arah Minor						
G5 atas + G10 bawah	searah jarum jam	533,5106	1495,1728	1495,172	5873,14018	0.2545
G5 bawah + G10 atas	berlawanan jarum jam	2228,253	585,7443	1642,508	5873,1401	0.2796

Contoh perhitungan

Vu kolom didapatkan dari kapasitas balok

C + T = 1339.673+1288.543 = 2628.217 kN

Vu kolom = (1018.407+1234.873)/3 = 751.0933 kN

(3 didapatkan dari rata-rata kolom atas dan bawah joint (1.5+4.5)/2=3)

Vuj = Cu + T - Vu kolom = 2628.218-751.0933 = 1877.123

Kapasitas joint = $1.0925\ x\ 0.85\ x\ \sqrt{fc'}\ x1000$

Kapasitas joint = $1.0925\ x\ 0.85\ x\ \sqrt{40}\ x1000$

Kapasitas Joint = 5873.14 kN

D/C Ratio = Vuj/Kapasitas Joint = 1877.123/5873.14 = 0.2545

Tabel Rangkuman dari masing-masing desain struktur

Jumlah	Desain awal	SOS	SOS *post-processed*	SOS a*ssisted*
Tulangan kolom (kg)	72.817	77.716	78.962	68.010
Beton kolom (m³)	354,831	366,055	377,272	318,997
Tulangan balok (kg)	85.978	64.720	61.797	50.619
Beton balok (m³)	355,946	527,936	481,581	307,911
Total (Rp)	**Rp 2.702.836.756**	**Rp 2.781.738.749**	**Rp 2.721.419.548**	**Rp 2.185.102.500**