บทที่ 4

ผลการดำเนินงานวิจัย

4.1 บทนำ

ในบทนี้ จะกล่าวถึงผลของการดำเนินงาน ซึ่งจะแสดงผลของการวิเคราะห์โครงสร้างพื้นคอนกรีตเสริมเหล็ก ด้วยวิธีหน่วยแรงใช้งาน (working stress design) ด้วยโปรแกรม Mathematica 5.1 แล้วเปรียบเทียบกับการวิเคราะห์ด้วยมือ ตามทฤษฎี ในบทที่ 2 โดยจะมีตัวอย่างที่จะคำนวณอยู่ 2 ตัวอย่าง เพื่อเป็นการตรวจสอบว่าโครงสร้างที่เราได้ทำการวิเคราะห์นั้นได้รับ Input และประมวลผลอย่างถูกต้อง ในสองตัวอย่างนี้ เป็นการออกแบบพื้นคอนกรีตทางเดียว และพื้นคอนกรีตสองทาง

ตัวอย่างที่ 1 จะเป็นการวิเคราะห์ และออกแบบพื้นคอนกรีตเสริมเหล็กทางเดียว โดยในตัวอย่างนี้จะตรวจสอบโครงสร้าง โดยมีอาจารย์ที่ปรึกษาและกลุ่มผู้จัดทำร่วมวิเคราะห์ แล้วทำการเปรียบเทียบกับโปรแกรม Mathematica 5.1

ตัวอย่างที่ 2 จะเป็นการวิเคราะห์ และออกแบบพื้นคอนกรีตเสริมเหล็กสองทาง โดยจะเลือกการออกแบบด้วนวิธีที่ 2 ตามมาตรฐานของ ว.ส.ท. โดยในตัวอย่างนี้จะตรวจสอบโครงสร้าง โดยมีอาจารย์ที่ปรึกษาและกลุ่มผู้จัดทำร่วมวิเคราะห์ แล้วทำการเปรียบเทียบกับโปรแกรม Mathematica 5.1

4.2 การออกแบบพื้นทางเดียวด้วยโปรแกรม (ONE WAY SLAP DESIGN)

INPUT

Type of slap 1 = SIMPLY SUPPORTED TYPE

2 = ONE END CONTINUOUS

3 = BOTH ENDS CONTINUOUS

4 = CANTILEVER SLAP

Input =

Slap Number	S1	=
Type of slap	2	=
Slap shot size (Clear span length)	3	m
Slap long size (Clear span length)	7.25	m
Thickness assume	0.15	m
Covering	0.025	m
Live load	200	ksm
Finish load	50	ksm
fc'	144	ksc
Main steel quality	SR24	=
Temperature Steel quality	SR24	=
diameter main steel	9	mm
diameter temperature steel	9	mm

Slab Compute; Slab Output;

SLAP DATA

Slap Number	S1	
Type of slap	ONE END CONTINUOUS	
Slap shot size (Clear span length)	3	m

Slap long size (Clear span length)	**7.25**	**m**
Thickness assume	**0.15**	**m**
Covering	**0.025**	**m**
MATERIAL DATA		
fc'	**144**	**ksc**
fc	64.8	ksc
Main Steel quality	SR24	
fy	2400	ksc
fs	1200	ksc
Temperature Steel quality	SR24	
fy	2400	ksc
fs	1200	ksc
Diameter main steel	9	mm
Diamcter temperature steel	9	mm
LOAD DATA		
Live load	200	kg/m^2
Finishing load	50	kg/m^2

.....CALULATE MINIMUM THICKNESS.....

Minimum Thickness = l/24 = 0.125 m

.....CALULATE PARAMETER....

$$n = \frac{2140000}{15810\sqrt{fc'}} = 11$$

$$k = \frac{1}{1+\frac{fs}{n\,fc}} = 0.372647$$

$$j = 1-\frac{k}{3} = 0.875784$$

$$R = \frac{1}{2}\times fc\times j\times k = 10.874\ ksc$$

.....CALULATE Load....

Dead Load = 2400 x Thickness = 360. kg/m^2

Total Load HWL = Dead Load + Live Load + Finishing Load = 610. kg/m^2

.....CALULATE MOMENT AND SHEAR.....

$$M+ = \frac{w*l^2}{11} = 499.091 \text{ kg-m}$$

$$M\text{-discont} = \frac{w*l^2}{24} = 228.75 \text{ kg-m}$$

$$M\text{-cont} = \frac{w*l^2}{9} = 610. \text{ kg-m}$$

Mmax = 610. kg-m

$$Vmax = \frac{w*l}{2} = 915. \text{ kg}$$

.....CALULATE EFFECTIVE DEEP.....

$$\text{Minimum Effective Deep} = \sqrt{\frac{Mmax}{Rb}} = 7.5953 \text{ cm}$$

$$\text{Using Effective Deep} = t\text{-covering} - \frac{\text{diameter main steel}}{2} = 12.05 \text{ cm}$$

.....CALULATE MINIMUM STEEL AREA OF TEMPERATURE STEEL.....

As Temperature steel need = 0.0025bt = 3.75 cm2

.....CALULATE MOMENT IN STRIP, AREA STEEL, SPACING.....

STRIP	MOMENT(kg-m)	Mr(kg-m)	As(cm^2)	USE(cm^2)
Middle	499.091	1535.37	3.94107	3.94107
Continuous	610.	1535.37	4.81686	4.81686
Discontinuous	228.75	1535.37	1.80632	3.75
Temperature steel			3.75	3.75

.....CALULATE CONCRETE SHEAR RESISTANT.....

$$Vc = 0.29\sqrt{fc'}\,b*d = 4193.4 \text{ kg}$$

.....CALULATE BOND AND PERIMETER.....

$$u = \frac{1.615\sqrt{fc'}}{(\text{diameter main steel}/10)} = 21 \text{ ksc}$$

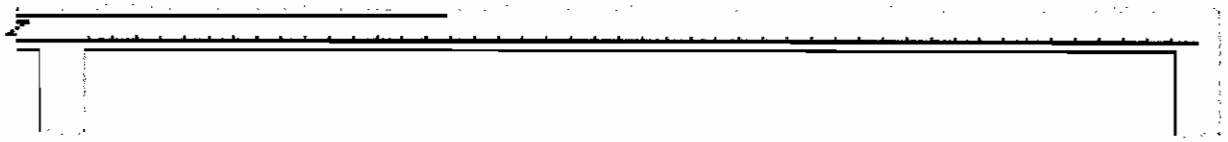
Perimeter Middle Strip = (π*Diameter main steel)*(1/Spacing) = 17.6715 cm

Perimeter Continuous Strip = (π*Diameter main steel)*(1/Spacing) = 21.7495 cm

Perimeter Discontinuous Strip = (π*Diameter main steel)*(1/Spacing) = 17.6715 cm

.....CALULATE LOAD ON BEAM.....

Load on beam = $\frac{w * l}{2}$ = 915. kg/m.



CALCULATION

```
SlabCompute :=
 Block[{},
  slapnum = Tinput[[1, 2]];
  tos = Tinput[[2, 2]];
  slapwide = Tinput[[4, 2]];
  l = Tinput[[3, 2]];
  tassume = Tinput[[5, 2]];
  covering = Tinput[[6, 2]];
  ll = Tinput[[7, 2]];
  fl = Tinput[[8, 2]];
  fcu = Tinput[[9, 2]];
  mainsq = Tinput[[10, 2]];
  tempsq = Tinput[[11, 2]];
  dmain = Tinput[[12, 2]];
  dtemp = Tinput[[13, 2]];
  If[mainsq === SR24,
{fymain = 2400 , fsmain = 1200, u = N[1.615 Sqrt[fcu] / (dmain / 10)], If[u ≥ 11, u = 11], nameu = " 1.615 Sqrt[fc'] / (diameter main steel / 10) = "},
If[mainsq === SD30, {fymain = 3000 , fsmain = 1500, u = N[3.35 Sqrt[fcu] / (dmain / 10)],
  If[u ≥ 35, u = 35], nameu = " 3.35 Sqrt[fc'] / (diameter main steel / 10) = "},
 If[mainsq == SD40, {fymain = 4000 , fsmain = 1700, u = N[3.35 Sqrt[fcu] / (dmain / 10)],
   If[u ≥ 35, u = 35], nameu = " 3.35 Sqrt[fc'] / (diameter main steel / 10) = "},
 If[mainsq === SD50, {fymain = 5000 , fsmain = 1700, u = N[3.35 Sqrt[fcu] / (dmain / 10)], If[u ≥ 35, u = 35],
 nameu = " 3.35 Sqrt[fc'] / (diameter main steel / 10) = "}, mainsq = " Select main steel quality "]]]];
```

```
If[tempsq === SR24, {fytemp = 2400 , fstemp = 1200, astemp = 0.0025 * 100 * (tassume * 100),
  nameastemp = "0.0025bt = "},
 If[tempsq === SD30, {fytemp = 3000 , fstemp = 1500, astemp = 0.0020 * 100 * (tassume * 100),
   nameastemp = "0.0020bt = "},
  If[tempsq === SD40, {fytemp = 4000 , fstemp = 1700, astemp = 0.0018 * 100 * (tassume * 100),
    nameastemp = "0.0018bt = "},
   If[tempsq === SD50, {fytemp = 5000 , fstemp = 1700, astemp = 0.0018 * 100 * (tassume * 100),
     nameastemp = "0.0018bt = "}, tempsq = "Select temperature steel quality"]]]];
fc = N[0.45 * fcu];
n = Round[2040000 / (15210 Sqrt[fcu])];
k = N[1 / (1 + (fsmain / (n * fc)))];
j = N[1 - k / 3];
R = N[0.5 * fc * j * k];
dl = N[2400 * tassume];
w = N[dl + ll + fl];
If[tos == 1, {tmin = N[1 / 20], nametmin = "1/20 = ", Mp = N[w * l^2 / 8], nameMp = "w * l^2 / 8 = ", Mnd = 0, nameMnd = "",
  Mnc = 0, nameMnc = "", Vmax = N[w * l / 2], nameVmax = "w * l / 2 = ", nametos = "SIMPLY SUPPORTED TYPE"},
 If[tos == 2, {tmin = N[1 / 24], nametmin = "1/24 = ", Mp = N[w * l^2 / 11], nameMp = "w * l^2 / 11 = ",
  Mnd = N[w * l^2 / 24], nameMnd = "w * l^2 / 24 = ", Mnc = N[w * l^2 / 9], nameMnc = "w * l^2 / 9 = ",
  Vmax = N[w * l / 2], nameVmax = "w * l / 2 = ", nametos = "ONE END CONTINUOUS"},
 If[tos == 3, {tmin = N[1 / 28], nametmin = "1/28 = ", Mp = N[w * l^2 / 16], nameMp = "w * l^2 / 16 = ",
  Mnd = 0, nameMnd = "", Mnc = N[w * l^2 / 11], nameMnc = "w * l^2 / 11 = ", Vmax = w * l / 2,
  nameVmax = "w * l / 2 = ", nametos = "BOTH ENDS CONTINUOUS"},
 If[tos == 4, {tmin = N[1 / 10], nametmin = "1/10 = ", Mp = 0, nameMp = "", Mnd = 0, nameMnd = "",
  Mnc = N[w * l^2 / 2], nameMnc = "w * l^2 / 2 = ", Vmax = N[w * l], nameVmax = "w * l = ",
    nametos = "CANTILEVER SLAP"}, tos = "Select Type Of Slap"]]]];
Mmax = Max[Mp, Mnd, Mnc];
d = N[(tassume * 100) - (covering * 100) - (dmain / 10) / 2];
dmin = N[Sqrt[(Mmax * 100) / (R * 100)]];
Mr = N[R * d^2];

Vc = N[(0.29 Sqrt[fcu]) * 100 * d];
asMp = N[Mp * 100 / (fsmain * j * d)];
```

```
asMnc = N[Mnc * 100 / (fsmain * j * d)];
asMnd = N[Mnd * 100 / (fsmain * j * d)];
RasMp = asMp;
RasMnc = asMnc;
RasMnd = asMnd;
If[asMp < astemp, asMp = Max[asMp, astemp]];
If[asMnc < astemp, asMnc = Max[asMnc, astemp]];
If[asMnd < astemp, asMnd = Max[asMnd, astemp]];
sMp = Min[N[1 / (asMp / (π * dmain^2 / 400))], N[(3 * (tassume * 100))], 0.3];
sMnc = Min[N[1 / (asMnc / (π * dmain^2 / 400))], N[(3 * (tassume * 100))], 0.3];
sMnd = Min[N[1 / (asMnd / (π * dmain^2 / 400))], N[(3 * (tassume * 100))], 0.3];
stemp = Min[N[1 / (astemp / (π * dtemp^2 / 400))], N[(3 * (tassume * 100))], 0.3];
sMp = N[Floor[sMp * 100] / 100];
sMnc = N[Floor[sMnc * 100] / 100];
sMnd = N[Floor[sMnd * 100] / 100];
stemp = N[Floor[stemp * 100] / 100];
p = N[Vmax / (u * j * d)];
pMp = N[(π * (dmain / 10)) * (1 / sMp)];
pMnc = N[(π * (dmain / 10)) * (1 / sMnc)];
pMnd = N[(π * (dmain / 10)) * (1 / sMnd)];
If[tos = 1, {asMnc = 0, asMnd = 0, sMnc = 0, sMnd = 0, pMnc = 0, pMnd = 0},
 If[tos = 3, {asMnd = 0, sMnd = 0, pMnd = 0}, If[tos = 4, {asMp = 0, asMnd = 0, sMp = 0, sMnd = 0, pMp = 0, pMnd = 0}]]]
```

- Module for graphics

- Editing

```
graphicSection[tos_, {slablength_, thk_, covering_},
  {{dmain_, smain_}, {dtemp_, stemp_}}, tscale_ : 1, {{dtopL_ : 0, stopL_ : 0}, {dtopR_ : 0, stopR_ : 0}}] :=
 Module[{},
  nSpace = Round[slablength / stemp - 0.5];
  (*-----------------------------------------------------*)
freeL = {{0, 0}, {0, thk}, {slablength, thk}};
freeR = {{slablength, 0}, {0, 0}};
simpleL = {{0, 0}, {0, -2 thk}, {-2 thk, -2 thk}, {-2 thk, thk}, {slablength, thk}};
simpleR = {{slablength + 2 thk, thk}, {slablength + 2 thk, -2 thk}, {slablength, -2 thk}, {slablength, 0}, {0, 0}};
fixedL = {{0, 0}, {0, -2 thk}, {-2 thk, -2 thk}, {-2 thk, 0}, {-3 thk, 0}, {-3 thk, thk / 3},
  {-3.5 thk, thk / 3}, {-2.5 thk, 2 thk / 3}, {-3 thk, 2 thk / 3}, {-3 thk, thk}, {slablength, thk}};
```

```
fixedR = {{slablength + 3 thk, thk}, {slablength + 3 thk, 2 thk / 3},
  {slablength + 3.5 thk, 2 thk / 3}, {slablength + 2.5 thk, thk / 3},
  {slablength + 3 thk, thk / 3}, {slablength + 3 thk, 0}, {slablength + 2 thk, 0},
  {slablength + 2 thk, -2 thk}, {slablength, -2 thk}, {slablength, 0}, {0, 0}};
(*-------------------------------------------------------*)
If[tos == 1, emb = {thk, thk}; envelop = Join[simpleL, simpleR],
 If[tos == 2, emb = {3 thk, thk}; envelop = Join[fixedL, simpleR],
  If[tos = 3, emb = {3 thk, 3 thk}; envelop = Join[fixedL, fixedR],
   If[tos == 4, emb = {3 thk, 0}; envelop = Join[fixedL, freeR]]]];
(*-------------------------------------------------------*)
coveringX = covering + dmain / 1000;
coveringY = covering + dtop / 1000;
lowersteel = Polygon[{{0 - emb[[1]], covering}, {slablength + emb[[2]], covering},
   {slablength + emb[[2]], coveringX}, {0 - emb[[1]], coveringX}, {0 - emb[[1]], covering}}];
uppersteelL = Polygon[{{0 - emb[[1]], thk - covering}, {0.33 slablength, thk - covering},
   {0.33 slablength, thk - coveringY},
   {0 - emb[[1]], thk - coveringY}, {0 - emb[[1]], thk - covering}}];
uppersteelR = Polygon[{{0.67 slablength, thk - covering},
   {slablength + emb[[2]], thk - covering}, {slablength + emb[[2]], thk - coveringY},
   {0.67 slablength, thk - coveringY}, {0.67 slablength, thk - covering}}];
uppersteel = Polygon[{{0 - emb[[1]], thk - covering},
   {slablength + emb[[2]], thk - covering}, {slablength + emb[[2]], thk - coveringX},
   {0 - emb[[1]], thk - coveringX}, {0 - emb[[1]], thk - covering}}];
(*-------------------------------------------------------*)

  If[tos == 1, Show[Graphics[
     {Line[envelop], lowersteel,
      Table[
       Circle[
        {0.5 * (slablength - nSpace * stemp) + i, coveringX + dtemp / 2000}, dtemp / 2000],
       {i, 0, slablength, stemp}]}],
    AspectRatio → Automatic, PlotRange -> All]];
  If[tos == 2, Show[Graphics[
     {Line[envelop], lowersteel, uppersteelL,
      Table[
       Circle[
        {0.5 * (slablength - nSpace * stemp) + i, coveringX + dtemp / 2000}, dtemp / 2000],
       {i, 0, slablength, stemp}]}],
    AspectRatio → Automatic, PlotRange -> All]];
  If[tos == 3, Show[Graphics[
     {Line[envelop], lowersteel, uppersteelL, uppersteelR,
      Table[
       Circle[
        {0.5 * (slablength - nSpace * stemp) + i, coveringX + dtemp / 2000}, dtemp / 2000],
       {i, 0, slablength, stemp}]}],
    AspectRatio → Automatic, PlotRange -> All]];
  If[tos == 4, Show[Graphics[
     {Line[envelop], uppersteel,
      Table[
```

```
    Circle[
     {0.5 * (slablength - nSpace * stemp) + i, (thk - coveringX) - dtemp / 2000}, dtemp / 2000],
    {i, 0, slablength, stemp}]}],
 AspectRatio → Automatic, PlotRange -> All]]]
```

Finished

```
inclined[{pt1_, pt2_}, thk_] :=
 Module[{},
  islope = Apply[ArcTan, pt2 - pt1];
  ilength = Sqrt[Dot[pt2 - pt1, pt2 - pt1]];
  inumber = Round[ilength / thk];
  Table[
   Line[
    {pt1 + i * thk {Cos[islope], Sin[islope]},
     pt1 + thk {-Sin[islope], Cos[islope]}
      + (i + 1) * thk {Cos[islope], Sin[islope]}}],
   {i, 0, inumber}]]

dash[{pt1_, pt2_}, thk_] :=
 Module[{},
  islope = Apply[ArcTan, pt2 - pt1];
  ilength = Sqrt[Dot[pt2 - pt1, pt2 - pt1]];
  {Dashing[{0.02, 0.02}],
   Line[
    {pt1 + thk {-Sin[islope], Cos[islope]},
     pt2 + thk {-Sin[islope], Cos[islope]}}]}]

tEdge[{pt1_, pt2_}, thk_, edgeCode_] :=
 If[edgeCode == 0, {},
  If[edgeCode == 1, dash[{pt1, pt2}, 0.5 thk],
   If[edgeCode == 2, inclined[{pt1, pt2}, thk]]]]

graphicPlan[tos_, slabwide_, slablength_] :=
 Module[{},
  If[tos == 1, vEdgecode = {1, 1}, If[tos == 2, vEdgecode = {2, 1}, If[tos == 3, vEdgecode = {2, 2},
     If[tos == 4, vEdgecode = {2, 0}]]]];
  thk = Min[slabwide, slablength] / 20;
  Show[Graphics[{
     Line[{{0, 0}, {0, slablength}, {slabwide, slablength}, {slabwide, 0}, {0, 0}}],
     tEdge[{{0, slablength}, {slabwide, slablength}}, thk, vEdgecode[[1]]],
     tEdge[{{slabwide, 0}, {0, 0}}, thk, vEdgecode[[2]]],
     Text[ToString[slabwide] <> " m.", {slabwide / 2, 0}, {0, -1}],
     Text[ToString[slablength] <> " m.", {0, slablength / 2}, {-1, 0}]},
    AspectRatio → Automatic, PlotRange → All]]]
```

Module for OUTPUT

```
SlabCutput :=
 Block[{},
  graphicPlan[tos, slapwide, l]
  Print[""];
  Print[TableForm[{{"SLAP DATA", "", ""}, {"Slap Number", slapnum, ""}, {"Type of slap", nametos, ""},
     {"Slap shot size(Clear span lengh) ", l, "m"}, {"Slap long size(Clear span lengh)", slapwide, "m"},
     {"Thickness assume", tassume, "m"}, {"Covering", covering, "m"}, {"", "", ""}, {"MATERIAL DATA", "", ""},
     {"fc'", fcu, "ksc"}, {"fc", fc, "ksc"}, {"Main Steel quality", mainsq, ""}, {"fy", fymain, "ksc"},
     {"fs", fsmain, "ksc"}, {"Temperature Steel quality", tempsq, ""}, {"fy", fytemp, "ksc"},
     {"fs", fstemp, "ksc"}, {"Diameter main steel", dmain, "mm"}, {"Diameter temperature steel", dtemp, "mm"},
     {"", "", ""}, {"LOAD DATA", "", ""}, {"Live load", ll, "ksm"}, {"Finishing load", fl, "ksm"}}]];
  Print[""];
  Print["....CALULATE MINIMUM THICKNESS....."];
  Print["Minimum Thickness = ", nametmin, tmin, " m"];
  If[tassume < tmin, Print["Thickness Assume less then Thickness min"]];
  Print["....CALULATE PARAMETER....."];
  Print["n = ", 2040000/(15210 Sqrt[fc']), " = ", n];
  If[n < 6, Print["n less then 6"]];
  Print["k = ", 1/(1 + fs/(n*fc)), " = ", k];
  Print["j = 1-", k/3, " = ", j];
  Print["R = ", 1/2, "*fc*j*k = ", R, " ksc"];
  Print["....CALULATE LOAD....."];
  Print["Dead Load = 2400*Thickness = ", dl, " ksm"];
  Print["Total Load(W) = Dead Load+Live Load+Finishing Load = ", w, " ksm"];
  Print["....CALULATE MOMENT AND SHEAR....."];
  Print["M+ = ", nameMp, Mp, " kg-m"];
  Print["M-discont = ", nameMnd, Mnd, " kg-m"];
  Print["M-cont = ", nameMnc, Mnc, " kg-m"];
  Print["Mmax = ", Mmax, " kg-m"];
  Print["Vmax = ", nameVmax, Vmax, " kg"];
  Print["....CALULATE EFFECTIVE DEEP....."];
  Print["Minimum Effective Deep = ", Sqrt[Mmax/Rb], " = ", dmin, " cm"];
  Print["Using Effective Deep = t-covering-", diameter main steel/2, " = ", d, " cm"];
  Print["....CALULATE MINIMUM STEEL AREA OF TEMPERATURE STEEL....."];
  Print["As Temperature steel need = ", nameastemp, astemp, " cm^2"];
```

```
If[dtemp < 6, Print["Diameter temperature steel less then 6 mm"]];
Print[".....CALULATE MOMENT IN STRIP,AREA STEEL,SPACING....."];
Print[""];
Print[TableForm[{{"STRIP", "MOMENT(kg-m)", "Mr(kg-m)", "As(cm²)", "USE(cm²)", "SPACING(m)"},
   {"Middle", Mp, Mr, RasMp, asMp, sMp}, {"Continuous", Mnc, Mr, RasMnc, asMnc, sMnc},
   {"Discontinuous", Mnd, Mr, RasMnd, asMnd, sMnd}, {"Temperature steel", "", "", astemp, astemp, stemp}}]];
Print[""];
If[Mr < Min[Mp, Mnc, Mnd], Print["M more then Mr increase thickness"]];
Print[".....CALULATE CONCRETE SHEAR RESISTANT....."];
Print["Vc= 0.29√fc' b*d = ", Vc, " kg"];
Print[".....CALULATE BOND AND PERIMETER....."];
Print["μ = ", nameu, u, " ksc"];
Print["Perimeter main steel need = Vmax/(μ*j*d) = ", p, " cm"];
Print["Perimeter Middle Strip = (π*Diameter main steel)*(1/Spacing) = ", pMp, " cm"];
Print["Perimeter Continuous Strip = (π*Diameter main steel)*(1/Spacing) = ", pMnc, " cm"];
Print["Perimeter Discontinuous Strip = (π*Diameter main steel)*(1/Spacing) = ", pMnd, " cm"];
If[p > Max[pMp, pMnc, pMnd], Print["Perimeter main steel is less"]];
Print[".....CALULATE LOAD ON BEAM....."];
Print["Load on beam = ", nameVmax, Vmax, " kg/m"];
Print[""];
graphicSection[tos, {slapwide, tassume, covering},
 {{dmain, sMp}, {dtemp, stemp}}, 1, {{dmain, sMnc}, {dtopR, stopR}}]
```

4.3 การออกแบบพื้นสองทางด้วยโปรแกรม (TWO WAY SLAP DESIGN)

INPUT

Type of slap

1 = CONTINUOUS ALL SIZE

2 = DISCONTINUOUS ONE SIZE

3 = DISCONTINUOUS TWO SIZE

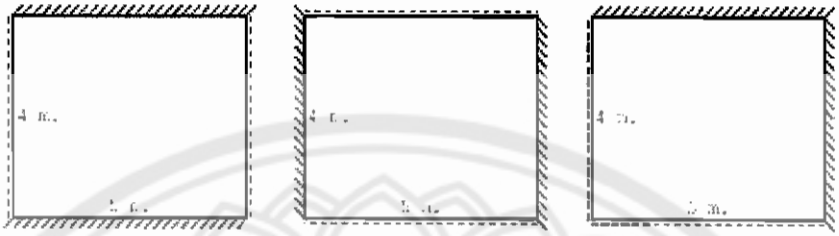
4 = DISCONTINUOUS THREE SIZE

5 = DISCONTINUOUS ALL SIZE

Tinput =

Slap Number	1	=
Type of slap	3	=
Slap shot size (Clear span lengh)	4	m
Slap long size (Clear span lengh)	5	m
Thickness assume	0.1	m
Covering	0.025	m
Live load	200	ksm
Finish load	50	ksm
fc'	200	ksc
steel quality	SD30	=
diameter steel	10	mm

Slab Compute; Slab Output;



SLAP DATA

Slap Number	1	
Type of slap	DISCONTINUOUS TWO SIZES	
Slap shot size clear span length	4	m
Slap long size clear span length	5	m
Thickness assume	0.1	m
Covering	0.025	m

MATERIAL DATA

fc'	200	ksc
fc	90.	ksc
Steel quality	SD30	
fy	3000	ksc
fs	1500	ksc
Diameter steel	10	mm

LOAD DATA

Live load	**200**	**kg/m^2**
Finishing load	**50**	**kg/m^2**

.....CALULATE MINIMUM THICKNESS.....

$$\text{Minimum Thickness} = \text{MAX}[(\frac{1}{180}\times(\text{Slap Perimeter}), 0.08\ \text{m}] = 0.1\ \text{m}$$

.....CALULATE PARAMETER.....

$$n = \frac{2040000}{15210\sqrt{fc'}} = 9$$

$$k = \frac{1}{1-\frac{fs}{(n*fc)}} = 0.350649$$

$$j = 1-\frac{k}{3} = 0.883117$$

$$R = \frac{1}{2}*fc*j*k = 13.9349 \text{ ksc}$$

.....CALULATE LOAD.....

Dead Load = 2400 x Thickness = 240. kg/m^2

Total Load HWL = Dead Load + Live Load + Finishing Load = 490. kg/m^2

.....CALULATE MOMENT AND SHEAR.....

$$m=\frac{S}{L} = 0.8$$

SHOT SPAN

STRIP	COEFFICIENT	M=CwS^2 (kg-m)
Continuous	0.084	501.76
Discontinuous	0.032	250.88
Middle	0.048	376.32

LONG SPAN

STRIP	COEFFICIENT	M=CwS^2 (kg-m)
Continuous	0.049	384.16
Discontinuous	0.025	196.
Middle	0.037	290.08

M max = 501.76 kg-m

V max = wl/ 2 = 1225. kg

.....CALULATE EFFECTIVE DEEP.....

$$\text{Minimum Effective Deep} = \sqrt{\frac{Mmax}{Rb}} = 6.00062 \text{ cm.}$$

$$\text{Using Effective Deep} = t\text{-covering}-\frac{\text{diameter main steel}}{2} = 7. \text{ cm.}$$

.....CALULATE MINIMUM STEEL AREA.....

Minimum Steel Area Require = 0.0020bt = 2. cm^2

.....CALULATE MOMENT IN STRIP, AREA STEEL, SPACING.....

SHOT SPAN

STRIP	MOMENT(kg-m)	Mr(kg-m)	As(cm^2)	USE(cm^2)	SPACING(m)
Continuous	501.76	682.81	5.41114	5.41114	0.14
Discontinuous	250.88	682.81	2.70557	2.70557	0.29
Middle	376.32	682.81	4.05835	4.05835	0.19

LONG SPAN

STRIP	MOMENT(kg-m)	Mr(kg-m)	As(cm^2)	USE(cm^2)	SPACING(m)
Continuous	384.16	682.81	4.1429	4.1429	0.18
Discontinuous	196.	682.81	2.11373	2.11373	0.3
Middle	290.08	682.81	3.12831	3.12831	0.25

.....CALULATE CONCRETE SHEAR RESISTANT.....

$Vc = 0.29\sqrt{f'c}\,b*d = 2870.88$ kg

.....CALULATE BOND AND PERIMETER.....

$$\mu = \frac{3.35\sqrt{f'c}}{(\text{diameter main steel}/10)} = 38 \text{ ksc}$$

$$\text{Perimeter main steel need} = \frac{Vmax}{\mu * j * d} = 5.66176 \text{ cm}$$

SHOT SPAN

STRIP	Perimeter Require(cm)	Using Perimeter(cm)
Middle	5.66176	16.5347
Continuous	5.66176	22.4399
Discontinuous	5.66176	10.8331

LONG SPAN

STRIP	Perimeter Require(cm)	Using Perimeter(cm)
Middle	5.66176	12.5664
Continuous	5.66176	17.4533
Discontinuous	5.66176	10.472

.....CALULATE LOAD ON BEAM.....

SHOT SPAN

Load On Beam Of Continuous Strip = $2[\frac{WS}{3}]$ = 1306.67 kg/m

Load On Beam Of Discontinuous Strip = $[\frac{WS}{3}]$ = 653.333 kg/m

LONG SPAN

Load On Beam Of Continuous Strip = $2[\frac{WS}{3}(\frac{3-m^2}{2})]$ = 1541.87 kg/m

Load On Beam Of Discontinuous Strip = $[\frac{WS}{3}(\frac{3-m^2}{2})]$ = 770.933 kg/m

CALCULATION

```
SlabCompute :=
 Block[{},
  slapnum = Tinput[[1, 2]];
  tos = Tinput[[2, 2]];
  shotsize = Tinput[[3, 2]];
  longsize = Tinput[[4, 2]];
  tassume = Tinput[[5, 2]];
  covering = Tinput[[6, 2]];
  ll = Tinput[[7, 2]];
  fl = Tinput[[8, 2]];
  fcu = Tinput[[9, 2]];
  sq = Tinput[[10, 2]];
  dmain = Tinput[[11, 2]];
  If[tos = 1, nametos = "CONTINUOUS ALL SIZE",
   If[tos = 2, nametos = "DISCONTINUOUS ONE SIZE",
    If[tos = 3, nametos = "DISCONTINUOUS TWO SIZE",
     If[tos = 4, nametos = "DISCONTINUOUS THREE SIZE",
      If[tos = 5, nametos = "DISCONTINUOUS ALL SIZE"]]]]];
```

$$\text{If}\left[sq === SR24,\ \left\{fy = 2400,\ fs = 1200,\ u = N\left[\frac{1.615\sqrt{fcu}}{(dmain/10)}\right],\ \text{If}[u \geq 11,\ u = 11],\right.\right.$$

$$\left.\left.nameu = "\frac{1.615\sqrt{fc'}}{(\text{diameter main steel}/10)} = ",\ asmin = 0.0025*100*(tassume*100),\ nameasmin = "0.0025bt = "\right\},\right.$$

$$\text{If}\left[sq === SD30,\ \left\{fy = 3000,\ fs = 1500,\ u = N\left[\frac{3.35\sqrt{fcu}}{(dmain/10)}\right],\ \text{If}[u \geq 35,\ u = 35],\right.\right.$$

$$\left.\left.nameu = "\frac{3.35\sqrt{fc'}}{(\text{diameter main steel}/10)} = ",\ asmin = 0.0020*100*(tassume*100),\ nameasmin = "0.0020bt = "\right\},\right.$$

$$\text{If}\left[sq == SD40,\ \left\{fy = 4000,\ fs = 1700,\ u = N\left[\frac{3.35\sqrt{fcu}}{(dmain/10)}\right],\ \text{If}[u \geq 35,\ u = 35],\right.\right.$$

```
nameu = "3.35√fc' / (diameter main steel / 10) = ", asmin = 0.0018*100*(tassume*100), nameasmin = "0.0018bt = "},

If[sq === SD50, {fy = 5000, fs = 1700, u = N[3.35√fcu / (dmain / 10)], If[u ≥ 35, u = 35],

nameu = "3.35√fc' / (diameter main steel / 10) = ", asmin = 0.0018*100*(tassume*100),
nameasmin = "0.0018bt = "}, sq = " Select main steel quality "]]]];

tmin = N[Max[(shotsize + longsize) / 90, 0.08]];
m = N[Ceiling[(shotsize / longsize)*100] / 100];
If[m < 0.5, m = 0.5];
fc = N[0.45*fcu];
n = Round[2040000 / (15210√fcu)];
k = N[1 / (1 + (fs / (n*fc)))];
j = N[1 - k / 3];
R = N[0.5*fc*j*k];
dl = N[2400*tassume];
w = N[dl + ll + fl];
(*.....long size coeficient.....*)

If[tos = 1, {CLMnc = 0.033, CLMnd = 0, CLMp = 0.025}, If[tos = 2, {CLMnc = 0.041, CLMnd = 0.021, CLMp = 0.031},
  If[tos = 3, {CLMnc = 0.049, CLMnd = 0.025, CLMp = 0.037}, If[tos = 4, {CLMnc = 0.058, CLMnd = 0.029, CLMp = 0.044},
    If[tos = 5, {CLMnc = 0, CLMnd = 0.033, CLMp = 0.050}]]]];
(*.....long size coeficient.....*)
(*.....shot size coeficient....*)
```

0.5`	0.083`	0.`	0.062`	0.5`	0.085`
0.51`	0.081`	0.`	0.0605`	0.51`	0.0834`
0.52`	0.079`	0.`	0.059`	0.52`	0.0818`
0.53`	0.077`	0.`	0.0575`	0.53`	0.08020000000000001`
0.54`	0.077`	0.`	0.0575`	0.54`	0.08020000000000001`
0.55`	0.073`	0.`	0.05449999999999999`	0.55`	0.077`
0.56`	0.071`	0.`	0.05299999999999999`	0.56`	0.0754`
0.57`	0.069`	0.`	0.051500000000000004`	0.57`	0.07380000000000002`
0.58`	0.067`	0.`	0.05`	0.58`	0.07220000000000001`
0.59`	0.065`	0.`	0.0485`	0.59`	0.07060000000000001`
0.6`	0.063`	0.`	0.047`	0.6`	0.069`
0.61`	0.0622`	0.`	0.0464`	0.61`	0.0683`
0.62`	0.061399999999999996`	0.`	0.0458`	0.62`	0.06760000000000001`
0.63`	0.0606`	0.`	0.0452`	0.63`	0.0669`
0.64`	0.0598`	0.`	0.0446`	0.64`	0.0662`
0.65`	0.059`	0.`	0.044`	0.65`	0.0655`

```
CS = {{0.66`, 0.058199999999999995`, 0.`, 0.0434`, 0.66`, 0.0648`},
      {0.67`, 0.05739999999999999`, 0.`, 0.0428`, 0.67`, 0.06409999999999999`},
      {0.68`, 0.0566`, 0.`, 0.042199999999999994`, 0.68`, 0.0634`},
      {0.69`, 0.0558`, 0.`, 0.041600000000000005`, 0.69`, 0.0627`},
      {0.7`, 0.055`, 0.`, 0.041`, 0.7`, 0.062`},
      {0.71`, 0.0543`, 0.`, 0.0405`, 0.71`, 0.0613`},
      {0.72`, 0.0536`, 0.`, 0.04`, 0.72`, 0.0606`},
      {0.73`, 0.052899999999999996`, 0.`, 0.0395`, 0.73`, 0.059899999999999995`},
      {0.74`, 0.052899999999999996`, 0.`, 0.0395`, 0.74`, 0.059199999999999996`},
      {0.75`, 0.052899999999999996`, 0.`, 0.0395`, 0.75`, 0.058499999999999996`},
      {0.76`, 0.0508`, 0.`, 0.038`, 0.76`, 0.058499999999999996`},
      {0.77`, 0.0501`, 0.`, 0.0375`, 0.77`, 0.058499999999999996`},
      {0.78`, 0.0494`, 0.`, 0.03699999999999999`, 0.78`, 0.05639999999999999`},
      {0.79`, 0.04869999999999999`, 0.`, 0.03649999999999999`, 0.79`, 0.05569999999999999`},
      {0.8`, 0.047999999999999994`, 0.`, 0.03599999999999999`, 0.8`, 0.055`},
      {0.81`, 0.0472`, 0.`, 0.035399999999999994`, 0.81`, 0.0543`},
      {0.82`, 0.04640000000000001`, 0.`, 0.034800000000000005`, 0.82`, 0.05360000000000001`},
      {0.83`, 0.04560000000000001`, 0.`, 0.0342`, 0.83`, 0.05290000000000001`},
      {0.84`, 0.04560000000000001`, 0.`, 0.0342`, 0.84`, 0.05290000000000001`},
      {0.85`, 0.044000000000000004`, 0.`, 0.033`, 0.85`, 0.05150000000000004`},
      {0.86`, 0.0432`, 0.`, 0.0324`, 0.86`, 0.05080000000000005`},
      {0.87`, 0.04240000000000001`, 0.`, 0.0318`, 0.87`, 0.05010000000000006`},
      {0.88`, 0.041600000000000005`, 0.`, 0.031200000000000002`, 0.88`, 0.04940000000000006`},
      {0.89`, 0.0408`, 0.`, 0.030600000000000002`, 0.89`, 0.0487`},
      {0.9`, 0.04`, 0.`, 0.03`, 0.9`, 0.048`},
      {0.91`, 0.0393`, 0.`, 0.0295`, 0.91`, 0.0473`},
      {0.92`, 0.0386`, 0.`, 0.028999999999999998`, 0.92`, 0.0466`},
      {0.93`, 0.037899999999999996`, 0.`, 0.028499999999999998`, 0.93`, 0.04589999999999996`},
      {0.94`, 0.037200000000000004`, 0.`, 0.028000000000000004`, 0.94`, 0.04520000000000004`},
      {0.95`, 0.036500000000000005`, 0.`, 0.0275`, 0.95`, 0.04450000000000005`},
      {0.96`, 0.035800000000000005`, 0.`, 0.027000000000000003`, 0.96`, 0.04380000000000006`},
      {0.97`, 0.035100000000000006`, 0.`, 0.026500000000000003`, 0.97`, 0.04310000000000006`},
      {0.98`, 0.03440000000000001`, 0.`, 0.026000000000000002`, 0.98`, 0.04240000000000001`},
      {0.99`, 0.0337`, 0.`, 0.025500000000000002`, 0.99`, 0.0417`},
      {1.`, 0.033`, 0.`, 0.025`, 1.`, 0.041`}}

(*.... ... shot size coeficient....*)
If[tos = 1, {i := 1, While[(CS[[i, 1]] ≠ m), i = i + 1], CSMnc = CS[[i, 2]], CSMnd = CS[[i, 3]], CSMp = CS[[i, 4]]}];
If[tos = 2, {i := 1, While[(CS[[i, 5]] ≠ m), i = i + 1], CSMnc = CS[[i, 6]], CSMnd = CS[[i, 7]], CSMp = CS[[i, 8]]}];
If[tos = 3,
 {i := 1, While[(CS[[i, 9]] ≠ m), i = i + 1], CSMnc = CS[[i, 10]], CSMnd = CS[[i, 11]], CSMp = CS[[i, 12]]}];
If[tos = 4, {i := 1, While[(CS[[i, 13]] ≠ m), i = i + 1], CSMnc = CS[[i, 14]],
  CSMnd = CS[[i, 15]], CSMp = CS[[i, 16]]}];
```

```
If[tos = 5, {i := 1, While[(CS[[i, 17]] ≠ m), i = i + 1], CSMnc = CS[[i, 18]],
  CSMnd = CS[[i, 19]], CSMp = CS[[i, 20]]}];
shotMnc = N[CSMnc * w * shotsize^2];
shotMnd = N[CSMnd * w * shotsize^2];
shotMp = N[CSMp * w * shotsize^2];
longMnc = N[CLMnc * w * shotsize^2];
longMnd = N[CLMnd * w * shotsize^2];
longMp = N[CLMp * w * shotsize^2];
Mmax = Max[shotMp, shotMnd, shotMnc, longMp, longMnd, longMnc];
Vmax = N[(w * longsize)/2];
d = N[(tassume * 100) - (covering * 100) - ((dmain / 10)/2)];
dmin = N[Sqrt[(Mmax * 100)/(R * 100)]];
Mr = N[R * d^2];
Vc = N[(0.29 Sqrt[fcu]) * 100 * d];
shotasMp = N[(shotMp * 100)/(fs * j * d)];
shotasMnc = N[(shotMnc * 100)/(fs * j * d)];
shotasMnd = N[(shotMnd * 100)/(fs * j * d)];
longasMp = N[(longMp * 100)/(fs * j * d)];
longasMnc = N[(longMnc * 100)/(fs * j * d)];
longasMnd = N[(longMnd * 100)/(fs * j * d)];
shotRasMp = shotasMp;
shotRasMnc = shotasMnc;
shotRasMnd = shotasMnd;
longRasMp = longasMp;
longRasMnc = longasMnc;
longRasMnd = longasMnd;
If[shotasMp < asmin, shotasMp = Max[shotasMp, asmin]];
If[shotasMnc < asmin, shotasMnc = Max[shotasMnc, asmin]];
If[shotasMnd < asmin, shotasMnd = Max[shotasMnd, asmin]];
```

```
If[longasMp < asmin, longasMp = Max[longasMp, asmin]];
If[longasMnc < asmin, longasMnc = Max[longasMnc, asmin]];
If[longasMnd < asmin, longasMnd = Max[longasMnd, asmin]];
```

$$\text{shotsMp} = \text{Min}\left[N\left[1 \Big/ \left(\text{shotasMp} \Big/ \left(\frac{\pi * \text{dmain}^2}{400}\right)\right)\right], N[(3 * (\text{tassume} * 100))], 0.3\right];$$

$$\text{shotsMnc} = \text{Min}\left[N\left[1 \Big/ \left(\text{shotasMnc} \Big/ \left(\frac{\pi * \text{dmain}^2}{400}\right)\right)\right], N[(3 * (\text{tassume} * 100))], 0.3\right];$$

$$\text{shotsMnd} = \text{Min}\left[N\left[1 \Big/ \left(\text{shotasMnd} \Big/ \left(\frac{\pi * \text{dmain}^2}{400}\right)\right)\right], N[(3 * (\text{tassume} * 100))], 0.3\right];$$

$$\text{longsMp} = \text{Min}\left[N\left[1 \Big/ \left(\text{longasMp} \Big/ \left(\frac{\pi * \text{dmain}^2}{400}\right)\right)\right], N[(3 * (\text{tassume} * 100))], 0.3\right];$$

$$\text{longsMnc} = \text{Min}\left[N\left[1 \Big/ \left(\text{longasMnc} \Big/ \left(\frac{\pi * \text{dmain}^2}{400}\right)\right)\right], N[(3 * (\text{tassume} * 100))], 0.3\right];$$

$$\text{longsMnd} = \text{Min}\left[N\left[1 \Big/ \left(\text{longasMnd} \Big/ \left(\frac{\pi * \text{dmain}^2}{400}\right)\right)\right], N[(3 * (\text{tassume} * 100))], 0.3\right];$$

```
shotsMp = N[Floor[shotsMp * 100] / 100];
shotsMnc = N[Floor[shotsMnc * 100] / 100];
shotsMnd = N[Floor[shotsMnd * 100] / 100];
longsMp = N[Floor[longsMp * 100] / 100];
longsMnc = N[Floor[longsMnc * 100] / 100];
longsMnd = N[Floor[longsMnd * 100] / 100];
```

$$p = N\left[\frac{\text{Vmax}}{(u * j * d)}\right];$$

```
shotpMp = N[(π * (dmain / 10)) * (1 / shotsMp)];
shotpMnc = N[(π * (dmain / 10)) * (1 / shotsMnc)];
shotpMnd = N[(π * (dmain / 10)) * (1 / shotsMnd)];
longpMp = N[(π * (dmain / 10)) * (1 / longsMp)];
longpMnc = N[(π * (dmain / 10)) * (1 / longsMnc)];
longpMnd = N[(π * (dmain / 10)) * (1 / longsMnd)];
If[shotMp == 0, {shotasMp = 0, shotsMp = 0, shotpMp = 0}];
If[shotMnc == 0, {shotasMnc = 0, shotsMnc = 0, shotpMnc = 0}];
If[shotMnd == 0, {shotasMnd = 0, shotsMnd = 0, shotpMnd = 0}];
If[longMp == 0, {longasMp = 0, longsMp = 0, longpMp = 0}];
If[longMnc == 0, {longasMnc = 0, longsMnc = 0, longpMnc = 0}];
If[longMnd == 0, {longasMnd = 0, longsMnd = 0, longpMnd = 0}];
```

$$\text{otLc} = N\left[2 * \left(\frac{w * \text{shotsize}}{3}\right)\right];$$

$$\text{shotLd} = N\left[\left(\frac{w * \text{shotsize}}{3}\right)\right];$$

$$\text{longLc} = N\left[2 * \left(\left(\frac{w * \text{shotsize}}{3}\right) * \left(\frac{3 - m^2}{2}\right)\right)\right];$$

$$\text{longLd} = N\left[\left(\left(\frac{w * \text{shotsize}}{3}\right) * \left(\frac{3 - m^2}{2}\right)\right)\right];$$

Module for OUTPUT

```
SlabOutput :=
 Block[{},
  "------------------------------------------------------------";
  If[tos = 1,
   Show[GraphicsArray[{graphicPlan2[{2, 2, 2, 2}, longsize, shotsize]}], ImageSize → 160]];
  If[tos = 2,
   Show[GraphicsArray[{graphicPlan2[{2, 2, 2, 1}, longsize, shotsize],
      graphicPlan2[{2, 2, 1, 2}, longsize, shotsize]}], ImageSize → 330]];
  If[tos = 3,
   Show[GraphicsArray[
     {graphicPlan2[{2, 1, 2, 1}, longsize, shotsize], graphicPlan2[{1, 2, 1, 2}, longsize, shotsize],
      graphicPlan2[{2, 2, 1, 1}, longsize, shotsize]}], ImageSize → 500]];
  If[tos = 4,
   Show[GraphicsArray[{graphicPlan2[{1, 1, 1, 2}, longsize, shotsize],
      graphicPlan2[{1, 1, 2, 1}, longsize, shotsize]}], ImageSize → 330]];
  If[tos = 5,
   Show[GraphicsArray[{graphicPlan2[{1, 1, 1, 1}, longsize, shotsize]}], ImageSize → 160]];
  "------------------------------------------------------------";
  Print[""];
  Print[TableForm[{{"SLAP DATA", "", ""}, {"Slap Number", slapnum, ""},
     {"Type of slap", nametos, ""}, {"Slap shot size(Clear span lengh) ", shotsize, "m"},
     {"Slap long size(Clear span lengh)", longsize, "m"}, {"Thickness assume", tassume, "m"},
     {"Covering", covering, "m"}, {"", "", ""}, {"MATERIAL DATA", "", ""}, {"fc'", fcu, "ksc"}, {"fc", fc, "ksc"},
     {"Steel quality", sq, ""}, {"fy", fy, "ksc"}, {"fs", fs, "ksc"}, {"Diameter steel", dmain, "mm"},
     {"", "", ""}, {"LOAD DATA", "", ""}, {"Live load", ll, "ksm"}, {"Finishing load", fl, "ksm"}}]];
  Print[""];
  Print[".....CALULATE MINIMUM THICKNESS....."];
```

```
Print["n = 2040000/(15210 Sqrt[fc']) = ", n];
If[n < 6, Print["n less then 6"]];
Print["k = 1/(1 + fs/(n*fc)) = ", k];
Print["j = 1-k/3 = ", j];
Print["R = 1/2*fc*j*k = ", R, " ksc"];
Print[".....CALULATE LOAD....."];
Print["Dead Load = 2400*Thickness = ", dl, " ksm"];
Print["Total Load(W) = Dead Load+Live Load+Finishing Load = ", w, " ksm"];
Print[".....CALULATE MOMENT AND SHEAR....."];
Print["m=S/L = ", m];
Print[UnderBar["SHOT SPAN"]];
Print[TableForm[{{"STRIP", "COEFICIENT", "M=CwS^2(kg-m)"},
   {"Continuous", CSMnc, shotMnc}, {"Discontinuous", CSMnd, shotMnd}, {"Middle", CSMp, shotMp}}]];
Print[UnderBar["LONG SPAN"]];
Print[TableForm[{{"STRIP", "COEFICIENT", "M=CwS^2(kg-m)"},
   {"Continuous", CLMnc, longMnc}, {"Discontinuous", CLMnd, longMnd}, {"Middle", CLMp, longMp}}]];
Print["Mmax = ", Mmax, " kg-m"];
Print["Vmax = ", "wl/2 = ", Vmax, " kg"];
Print[".....CALULATE EFFECTIVE DEEP....."];
Print["Minimum Effective Deep = Sqrt[Mmax/Rb] = ", dmin, " cm"];
Print["Using Effective Deep = t-covering-diameter main steel/2 = ", d, " cm"];
Print[".....CALULATE MINIMUM STEEL AREA ....."];
Print["Minimum Steel Area Require = ", nameasmin, asmin, " cm^2"];
If[dmain < 6, Print["Diameter of steel less then 6 mm"]];
Print[".....CALULATE MOMENT IN STRIP,AREA STEEL,SPACING....."];
Print[UnderBar["SHOT SPAN"]];
Print[TableForm[{{"STRIP", "MOMENT(kg-m)", "Mr(kg-m)", "As(cm^2)", "USE(cm^2)", "SPACING(m)"},
   {"Continuous", shotMnc, Mr, shotRasMnc, shotasMnc, shotsMnc},
   {"Discontinuous", shotMnd, Mr, shotRasMnd, shotasMnd, shotsMnd},
   {"Middle", shotMp, Mr, shotRasMp, shotasMp, shotsMp}}]];
Print[UnderBar["LONG SPAN"]];
```

```
Print[TableForm[{{"STRIP", "MOMENT(kg-m)", "Mr(kg-m)", "As(cm²)", "USE(cm²)", "SPACING(m)"},
   {"Continuous", longMnc, Mr, longRasMnc, longasMnc, longsMnc},
   {"Discontinuous", longMnd, Mr, longRasMnd, longasMnd, longsMnd},
   {"Middle", longMp, Mr, longRasMp, longasMp, longsMp}}]];
If[Mr < Min[shotMp, shotMnc, shotMnd, longMp, longMnc, longMnd], Print["M more then Mr increase thickness"]];
Print["....CALULATE CONCRETE SHEAR RESISTANT....."];
Print["Vc= 0.29√fc' b*d = ", Vc, " kg"];
If[Vc < Vmax, Print["Vmax more then Vc increase thickness"]];
Print["....CALULATE BOND AND PERIMETER....."];
Print["μ = ", nameu, u, " ksc"];
Print["Perimeter main steel need = Vmax/(μ*j*d) = ", p, " cm"];
Print[UnderBar["SHOT SPAN"]];
Print[TableForm[{{"STRIP", "Perimeter Require(cm)", "Using Perimeter(cm)"},
   {"Middle", p, shotpMp}, {"Continuous", p, shotpMnc}, {"Discontinuous", p, shotpMnd}}]];
If[p > Max[shotpMp, shotpMnc, shotpMnd], Print["Perimeter main steel is less"]];
Print[UnderBar["LONG SPAN"]];
Print[TableForm[{{"STRIP", "Perimeter Require(cm)", "Using Perimeter(cm)"},
   {"Middle", p, longpMp}, {"Continuous", p, longpMnc}, {"Discontinuous", p, longpMnd}}]];
If[p > Max[shotpMp, shotpMnc, shotpMnd, longpMp, longpMnc, longpMnd], Print["Perimeter main steel is less"]];
Print["....CALULATE LOAD ON BEAM....."];
Print[UnderBar["SHOT SPAN"]];
Print["Load On Beam Of Continuous Strip = 2[wS/3] = ", shotLc, " kg/m"];
Print["Load On Beam Of Discontinuous Strip = [wS/3] = ", shotLd, " kg/m"];
Print[UnderBar["LONG SPAN"]];
Print["Load On Beam Of Continuous Strip = 2[(wS/3)((3-m²)/2)] = ", longLc, " kg/m"];
Print["Load On Beam Of Discontinuous Strip = [(wS/3)((3-m²)/2)] = ", longLd, " kg/m"];
Print[""]];
```

Finished

```
inclined[{pt1_, pt2_}, thk_] :=
 Module[{},
  islope = Apply[ArcTan, pt2 - pt1];
  ilength = Sqrt[Dot[pt2 - pt1, pt2 - pt1]];
  inumber = Round[ilength / thk];
```

```
  Table[
   Line[
    {pt1 + i * thk {Cos[islope], Sin[islope]},
     pt1 + thk {-Sin[islope], Cos[islope]}
      + (i + 1) * thk {Cos[islope], Sin[islope]}}],
   {i, 0, inumber}]]

dash[{pt1_, pt2_}, thk_] :=
 Module[{},
  islope = Apply[ArcTan, pt2 - pt1];
  ilength = Sqrt[Dot[pt2 - pt1, pt2 - pt1]];
  {Dashing[{0.02, 0.02}],
   Line[
    {pt1 + thk {-Sin[islope], Cos[islope]},
     pt2 + thk {-Sin[islope], Cos[islope]}}]}]

tEdge[{pt1_, pt2_}, thk_, edgeCode_] :=
 If[edgeCode == 0, {},
  If[edgeCode = 1, dash[{pt1, pt2}, 0.5 thk],
   If[edgeCode = 2, inclined[{pt1, pt2}, thk]]]]

graphicPlan[tos_, slabwide_, slablength_] :=
 Module[{},
  If[tos = 1, vEdgecode = {1, 1}, If[tos = 2, vEdgecode = {2, 1}, If[tos = 3, vEdgecode = {2, 2},
    If[tos = 4, vEdgecode = {2, 0}]]]];
  thk = Min[slabwide, slablength] / 20;
  Show[Graphics[{
    Line[{{0, 0}, {0, slablength}, {slabwide, slablength}, {slabwide, 0}, {0, 0}}],
    tEdge[{{0, slablength}, {slabwide, slablength}}, thk, vEdgecode[[1]]],
    tEdge[{{slabwide, 0}, {0, 0}}, thk, vEdgecode[[2]]],
    Text[ToString[slabwide] <> " m.", {slabwide / 2, 0}, {0, -1}],
    Text[ToString[slablength] <> " m.", {0, slablength / 2}, {-1, 0}]},
   AspectRatio → Automatic, PlotRange → All]]]

 graphicPlan2[vEdgecode_, slabwide_, slablength_] :=
  Module[{},
   thk = Min[slabwide, slablength] / 20;
   Graphics[{
     Line[{{0, 0}, {0, slablength}, {slabwide, slablength}, {slabwide, 0}, {0, 0}}],
     tEdge[{{0, slablength}, {slabwide, slablength}}, thk, vEdgecode[[1]]],
```

```
 tEdge[{{slabwide, slablength}, {slabwide, 0}}, thk, vEdgecode[[2]]],
 tEdge[{{slabwide, 0}, {0, 0}}, thk, vEdgecode[[3]]],
 tEdge[{{0, 0}, {0, slablength}}, thk, vEdgecode[[4]]],
 Text[ToString[slabwide] <> " m.", {slabwide / 2, 0}, {0, -1}],
 Text[ToString[slablength] <> " m.", {0, slablength / 2}, {-1, 0}]},
AspectRatio → Automatic, PlotRange → All]]
```