

**Laporan Pembuatan Objek
Grafik Komputer**



Disusun oleh:

C14200174 - Fernando Wahyu

C14200193 - Ramadhan Daffa

C14200195 - Audrico

Pengajar :

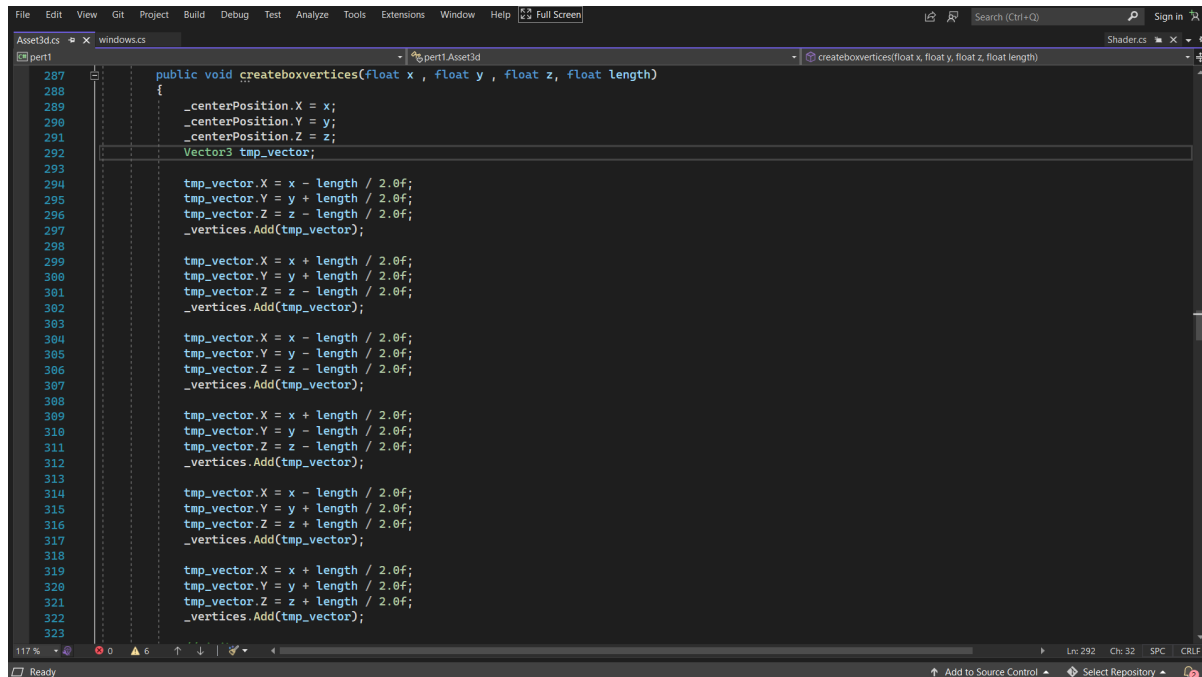
LILIANA, S.T.,M.ENG., PH.D

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS KRISTEN PETRA
SURABAYA**

A.Fungsi-Fungsi Pembangun

1. createboxvertices

Fungsi ini digunakan untuk membuat sebuah kubus. Dimana menggunakan 4 Parameter koordinat x y z serta lenght dari kubus dapat dilihat di bawah

A screenshot of the Visual Studio Code editor showing a C# script named Asset3d.cs. The script defines a public void function called createboxvertices that takes four float parameters: x, y, z, and length. The function sets the _centerPosition to (x, y, z) and then iteratively calculates and adds the vertices of a cube to the _vertices array. The vertices are calculated by adding and subtracting half the length from the center position coordinates. The code is as follows:

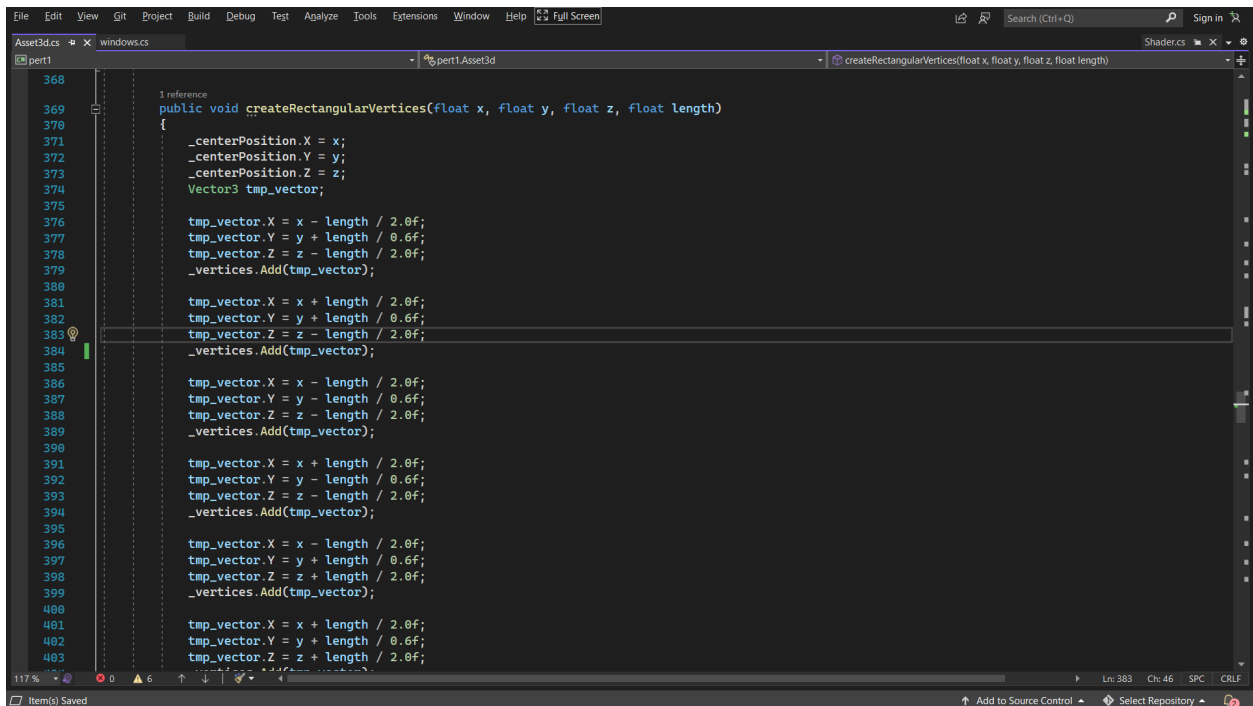
```
287 public void createboxvertices(float x , float y , float z, float length)
288 {
289     _centerPosition.X = x;
290     _centerPosition.Y = y;
291     _centerPosition.Z = z;
292     Vector3 tmp_vector;
293
294     tmp_vector.X = x - length / 2.0f;
295     tmp_vector.Y = y + length / 2.0f;
296     tmp_vector.Z = z - length / 2.0f;
297     _vertices.Add(tmp_vector);
298
299     tmp_vector.X = x + length / 2.0f;
300     tmp_vector.Y = y + length / 2.0f;
301     tmp_vector.Z = z - length / 2.0f;
302     _vertices.Add(tmp_vector);
303
304     tmp_vector.X = x - length / 2.0f;
305     tmp_vector.Y = y - length / 2.0f;
306     tmp_vector.Z = z - length / 2.0f;
307     _vertices.Add(tmp_vector);
308
309     tmp_vector.X = x + length / 2.0f;
310     tmp_vector.Y = y - length / 2.0f;
311     tmp_vector.Z = z - length / 2.0f;
312     _vertices.Add(tmp_vector);
313
314     tmp_vector.X = x - length / 2.0f;
315     tmp_vector.Y = y + length / 2.0f;
316     tmp_vector.Z = z + length / 2.0f;
317     _vertices.Add(tmp_vector);
318
319     tmp_vector.X = x + length / 2.0f;
320     tmp_vector.Y = y + length / 2.0f;
321     tmp_vector.Z = z + length / 2.0f;
322     _vertices.Add(tmp_vector);
323 }
```

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Full Screen
Asset3d.cs x windows.cs Shader.cs x x
pert1 createboxvertices(float x, float y, float z, float length)
317         _vertices.Add(tmp_vector);
318
319         tmp_vector.X = x + length / 2.0f;
320         tmp_vector.Y = y + length / 2.0f;
321         tmp_vector.Z = z + length / 2.0f;
322         _vertices.Add(tmp_vector);
323
324         //titik 7
325         tmp_vector.X = x - length / 2.0f;
326         tmp_vector.Y = y - length / 2.0f;
327         tmp_vector.Z = z + length / 2.0f;
328         _vertices.Add(tmp_vector);
329         //titik 8
330         tmp_vector.X = x + length / 2.0f;
331         tmp_vector.Y = y - length / 2.0f;
332         tmp_vector.Z = z + length / 2.0f;
333         _vertices.Add(tmp_vector);
334
335         _indices = new List<uint>
336         {
337             //SEGITIGA DEPAN 1
338             0,1,2,
339             //SEGITIGA DEPAN 2
340             1,2,3,
341             //SEGITIGA ATAS 1
342             0,4,5,
343             //SEGITIGA ATAS 2
344             0,1,5,
345             //SEGITIGA KANAN 1
346             1,3,5,
347             //SEGITIGA KANAN 2
348             3,5,7,
349             //SEGITIGA KIRI 1
350             0,2,4,
351             //SEGITIGA KIRI 2
352             2,4,6,
353             //SEGITIGA BELAKANG 1
```

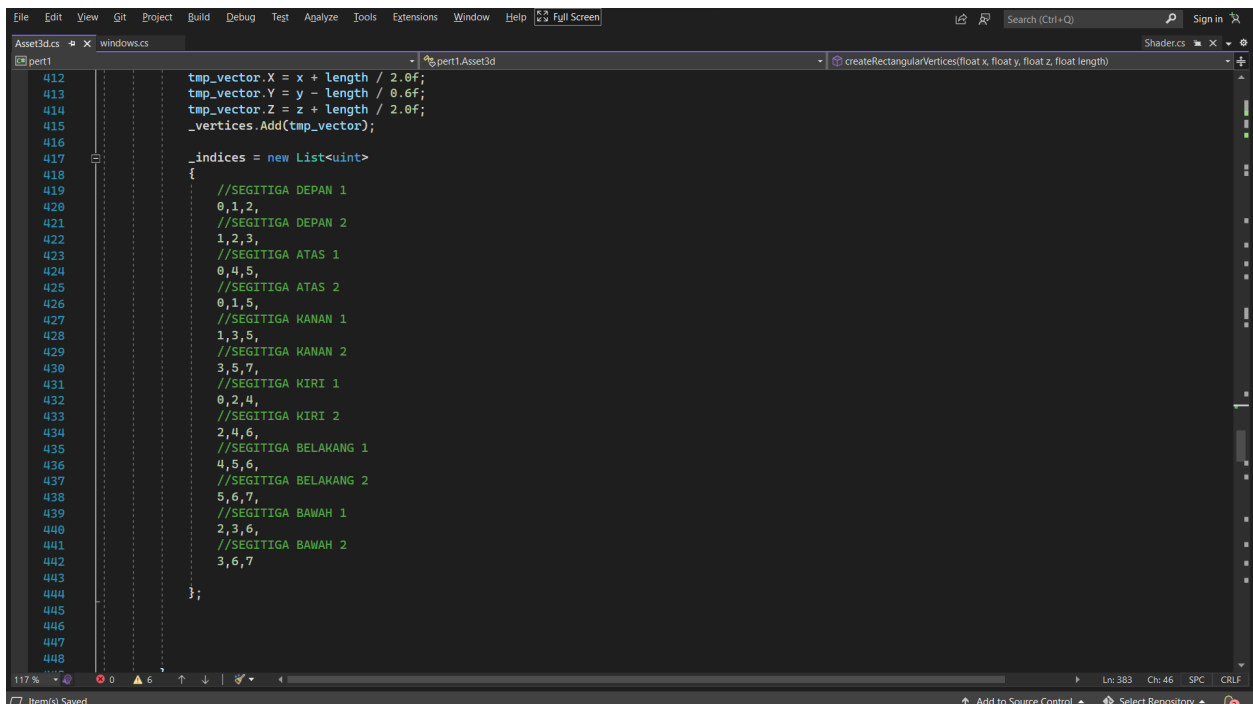
```
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Asset3d.cs x windows.cs Shader.cs x
per1 Asset3d createboxvertices(float x, float y, float z, float length)
332 tmp_vector.Z = z + length / 2.0f;
333 _vertices.Add(tmp_vector);
334
335 _indices = new List<uint>
336 {
337     //SEGITIGA DEPAN 1
338     0,1,2,
339     //SEGITIGA DEPAN 2
340     1,2,3,
341     //SEGITIGA ATAS 1
342     0,4,5,
343     //SEGITIGA ATAS 2
344     0,1,5,
345     //SEGITIGA KANAN 1
346     1,3,5,
347     //SEGITIGA KANAN 2
348     3,5,7,
349     //SEGITIGA KIRI 1
350     0,2,4,
351     //SEGITIGA KIRI 2
352     2,4,6,
353     //SEGITIGA BELAKANG 1
354     4,5,6,
355     //SEGITIGA BELAKANG 2
356     5,6,7,
357     //SEGITIGA BAWAH 1
358     2,3,6,
359     //SEGITIGA BAWAH 2
360     3,6,7
361 };
362
363
364
365
366
367
368
```

2. createRectangularVertices

Fungsi ini akan membuat bentuk balok, code ini merupakan modifikasi dari createBoxVertices().



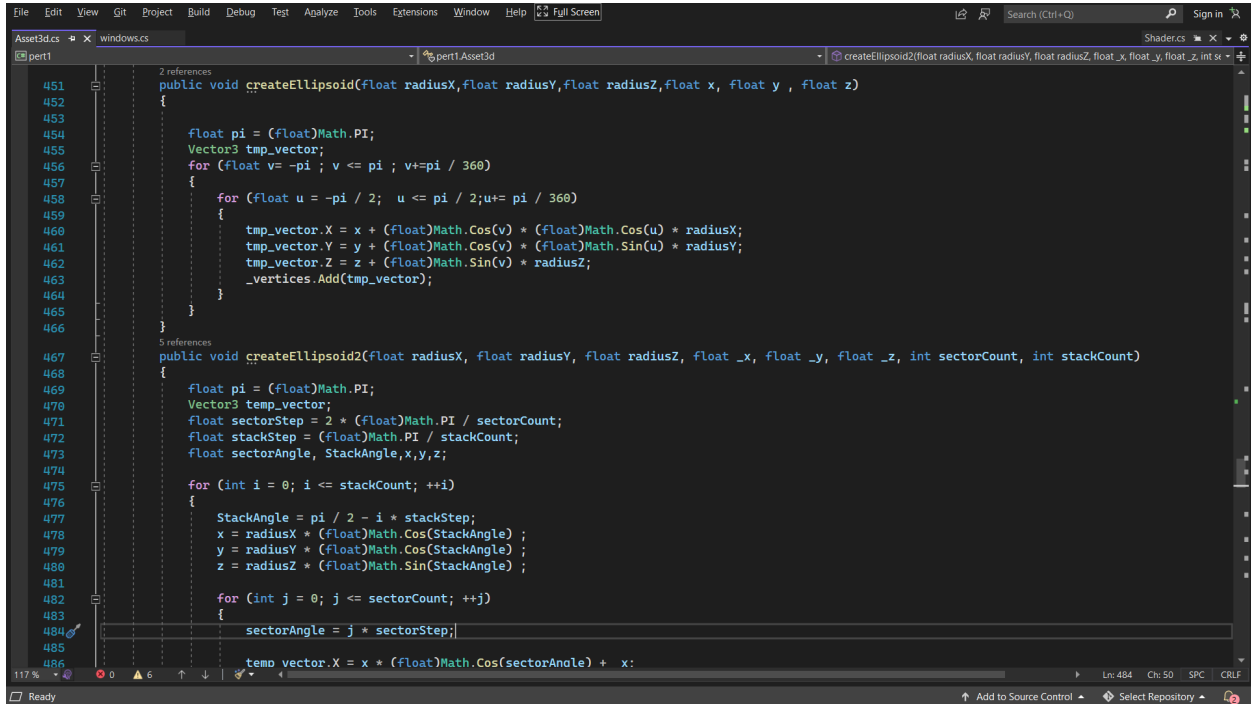
```
1 reference
369 public void createRectangularVertices(float x, float y, float z, float length)
370 {
371     _centerPosition.X = x;
372     _centerPosition.Y = y;
373     _centerPosition.Z = z;
374     Vector3 tmp_vector;
375
376     tmp_vector.X = x - length / 2.0f;
377     tmp_vector.Y = y + length / 0.6f;
378     tmp_vector.Z = z - length / 2.0f;
379     _vertices.Add(tmp_vector);
380
381     tmp_vector.X = x + length / 2.0f;
382     tmp_vector.Y = y + length / 0.6f;
383     tmp_vector.Z = z - length / 2.0f;
384     _vertices.Add(tmp_vector);
385
386     tmp_vector.X = x - length / 2.0f;
387     tmp_vector.Y = y - length / 0.6f;
388     tmp_vector.Z = z - length / 2.0f;
389     _vertices.Add(tmp_vector);
390
391     tmp_vector.X = x + length / 2.0f;
392     tmp_vector.Y = y - length / 0.6f;
393     tmp_vector.Z = z - length / 2.0f;
394     _vertices.Add(tmp_vector);
395
396     tmp_vector.X = x - length / 2.0f;
397     tmp_vector.Y = y + length / 0.6f;
398     tmp_vector.Z = z + length / 2.0f;
399     _vertices.Add(tmp_vector);
400
401     tmp_vector.X = x + length / 2.0f;
402     tmp_vector.Y = y + length / 0.6f;
403     tmp_vector.Z = z + length / 2.0f;
```



```
412     tmp_vector.X = x + length / 2.0f;
413     tmp_vector.Y = y - length / 0.6f;
414     tmp_vector.Z = z + length / 2.0f;
415     _vertices.Add(tmp_vector);
416
417     _indices = new List<uint>
418     {
419         //SEGITIGA DEPAN 1
420         0,1,2,
421         //SEGITIGA DEPAN 2
422         1,2,3,
423         //SEGITIGA ATAS 1
424         0,4,5,
425         //SEGITIGA ATAS 2
426         0,1,5,
427         //SEGITIGA KANAN 1
428         1,3,5,
429         //SEGITIGA KANAN 2
430         3,5,7,
431         //SEGITIGA KIRI 1
432         0,2,4,
433         //SEGITIGA KIRI 2
434         2,4,6,
435         //SEGITIGA BELAKANG 1
436         4,5,6,
437         //SEGITIGA BELAKANG 2
438         5,6,7,
439         //SEGITIGA BAWAH 1
440         2,3,6,
441         //SEGITIGA BAWAH 2
442         3,6,7
443     };
444
445
446
447
448
```

3. createEllipsoid2 dan createEllipsoid

Fungsi ini digunakan untuk membuat bangun elipsoid, dengan parameter yang digunakan ada radiusX, radiusY, radiusZ, _x, _y, _z, sectorCount, stackCount. Yang dimana radius X, Y dan Z digunakan untuk menentukan radius dari masing2 x, y, atau z. Sedangkan _x,y dan z digunakan untuk mengatur koordinasi suatu bidang.



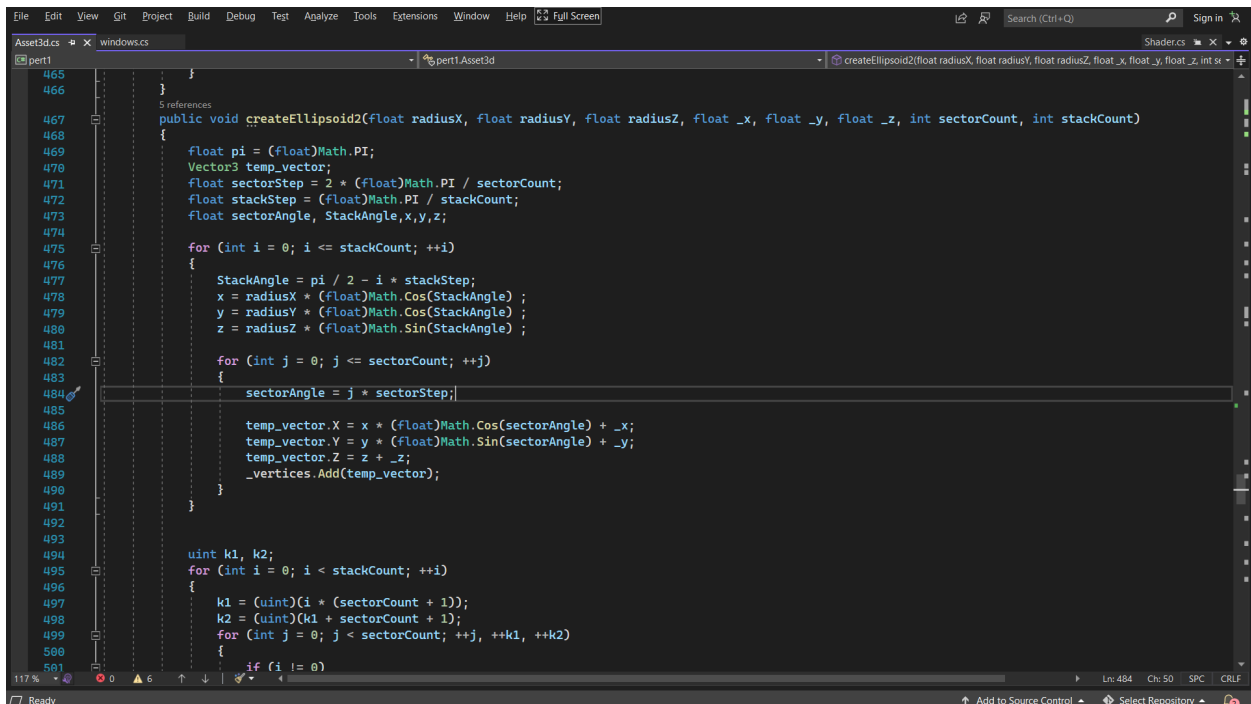
```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Sign in
Asset3dcs x windows.cs x %pert1 Asset3d createEllipsoid2(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z, int sectorCount, int stackCount) Shader.cs x
pert1
2 references
public void createEllipsoid(float radiusX, float radiusY, float radiusZ, float x, float y, float z)
{
    float pi = (float)Math.PI;
    Vector3 tmp_vector;
    for (float v = -pi; v <= pi; v+=pi / 360)
    {
        for (float u = -pi / 2; u <= pi / 2; u+= pi / 360)
        {
            tmp_vector.X = x + (float)Math.Cos(v) * (float)Math.Cos(u) * radiusX;
            tmp_vector.Y = y + (float)Math.Cos(v) * (float)Math.Sin(u) * radiusY;
            tmp_vector.Z = z + (float)Math.Sin(v) * radiusZ;
            _vertices.Add(tmp_vector);
        }
    }
}
5 references
public void createEllipsoid2(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z, int sectorCount, int stackCount)
{
    float pi = (float)Math.PI;
    Vector3 tmp_vector;
    float sectorStep = 2 * (float)Math.PI / sectorCount;
    float stackStep = (float)Math.PI / stackCount;
    float sectorAngle, StackAngle, x, y, z;

    for (int i = 0; i <= stackCount; ++i)
    {
        StackAngle = pi / 2 - i * stackStep;
        x = radiusX * (float)Math.Cos(StackAngle);
        y = radiusY * (float)Math.Cos(StackAngle);
        z = radiusZ * (float)Math.Sin(StackAngle);

        for (int j = 0; j <= sectorCount; ++j)
        {
            sectorAngle = j * sectorStep;

            tmp_vector.X = x * (float)Math.Cos(sectorAngle) + _x;
            tmp_vector.Y = y * (float)Math.Sin(sectorAngle) + _y;
            tmp_vector.Z = z + _z;
            _vertices.Add(tmp_vector);
        }
    }

    uint k1, k2;
    for (int i = 0; i < stackCount; ++i)
    {
        k1 = (uint)(i * (sectorCount + 1));
        k2 = (uint)(k1 + sectorCount + 1);
        for (int j = 0; j < sectorCount; ++j, ++k1, ++k2)
        {
            if (i != 0)
            {
                _vertices[k1] = _vertices[k2];
                _vertices[k2] = _vertices[k1 + 1];
                _vertices[k1 + 1] = _vertices[k2];
            }
        }
    }
}
```



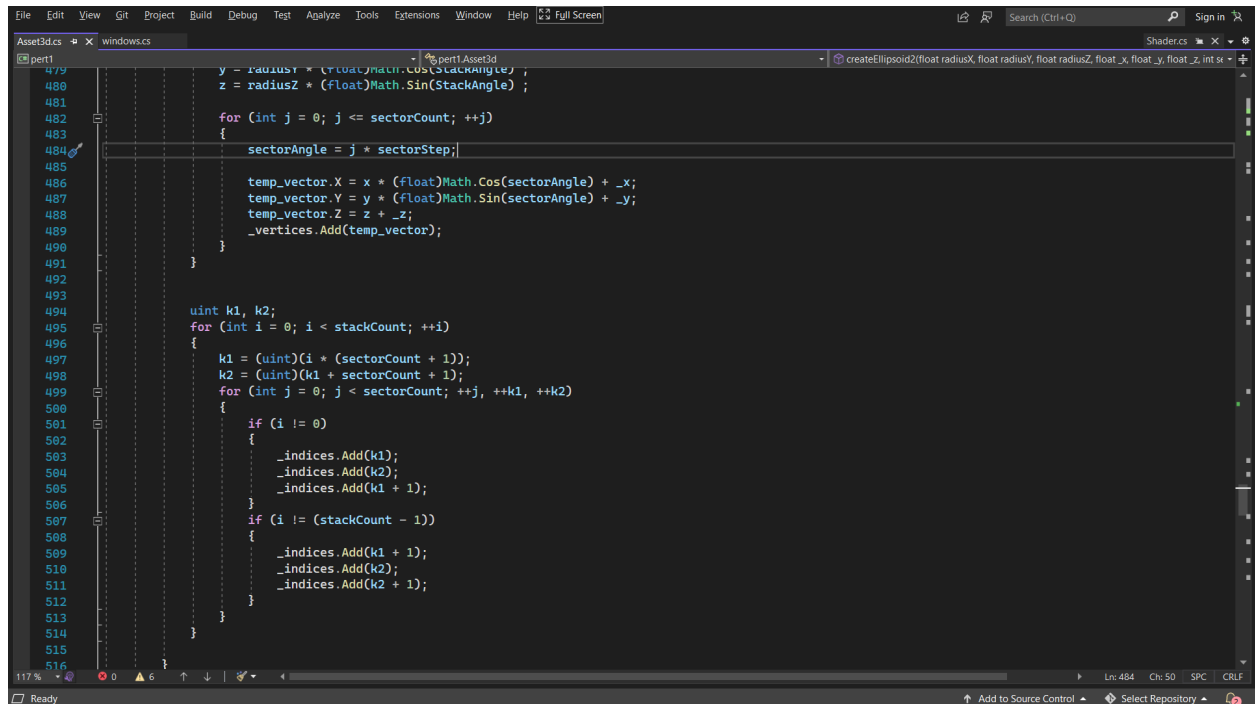
```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Sign in
Asset3dcs x windows.cs x %pert1 Asset3d createEllipsoid2(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z, int sectorCount, int stackCount) Shader.cs x
pert1
5 references
public void createEllipsoid2(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z, int sectorCount, int stackCount)
{
    float pi = (float)Math.PI;
    Vector3 tmp_vector;
    float sectorStep = 2 * (float)Math.PI / sectorCount;
    float stackStep = (float)Math.PI / stackCount;
    float sectorAngle, StackAngle, x, y, z;

    for (int i = 0; i <= stackCount; ++i)
    {
        StackAngle = pi / 2 - i * stackStep;
        x = radiusX * (float)Math.Cos(StackAngle);
        y = radiusY * (float)Math.Cos(StackAngle);
        z = radiusZ * (float)Math.Sin(StackAngle);

        for (int j = 0; j <= sectorCount; ++j)
        {
            sectorAngle = j * sectorStep;

            tmp_vector.X = x * (float)Math.Cos(sectorAngle) + _x;
            tmp_vector.Y = y * (float)Math.Sin(sectorAngle) + _y;
            tmp_vector.Z = z + _z;
            _vertices.Add(tmp_vector);
        }
    }

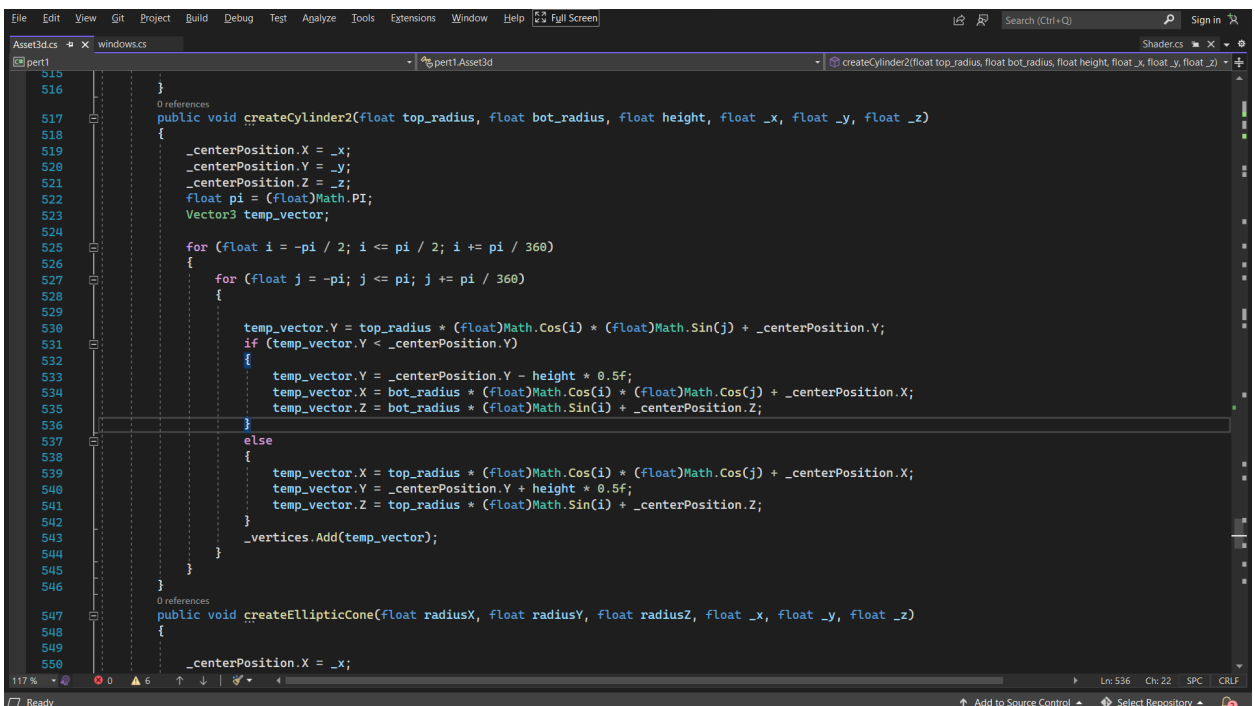
    uint k1, k2;
    for (int i = 0; i < stackCount; ++i)
    {
        k1 = (uint)(i * (sectorCount + 1));
        k2 = (uint)(k1 + sectorCount + 1);
        for (int j = 0; j < sectorCount; ++j, ++k1, ++k2)
        {
            if (i != 0)
            {
                _vertices[k1] = _vertices[k2];
                _vertices[k2] = _vertices[k1 + 1];
                _vertices[k1 + 1] = _vertices[k2];
            }
        }
    }
}
```



```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Full Screen
Asset3d.cs x windows.cs Shader.cs x
pert1 Asset3d createEllipsoid2(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z, int sectorCount, float _sectorStep)
479 y = radiusY * (float)Math.Cos(stackAngle);
480 z = radiusZ * (float)Math.Sin(stackAngle);
481
482 for (int j = 0; j <= sectorCount; ++j)
483 {
484     sectorAngle = j * sectorStep;
485
486     temp_vector.X = x * (float)Math.Cos(sectorAngle) + _x;
487     temp_vector.Y = y * (float)Math.Sin(sectorAngle) + _y;
488     temp_vector.Z = z + _z;
489     _vertices.Add(temp_vector);
490 }
491
492
493
494 uint k1, k2;
495 for (int i = 0; i < stackCount; ++i)
496 {
497     k1 = (uint)(i * (sectorCount + 1));
498     k2 = (uint)(k1 + sectorCount + 1);
499     for (int j = 0; j < sectorCount; ++j, ++k1, ++k2)
500     {
501         if (i != 0)
502         {
503             _indices.Add(k1);
504             _indices.Add(k2);
505             _indices.Add(k1 + 1);
506         }
507         if (i != (stackCount - 1))
508         {
509             _indices.Add(k1 + 1);
510             _indices.Add(k2);
511             _indices.Add(k2 + 1);
512         }
513     }
514 }
515
516
117% Lrn: 484 Ch: 50 SPC CRLF
Ready Add to Source Control Select Repository
```

4. createCylinder2

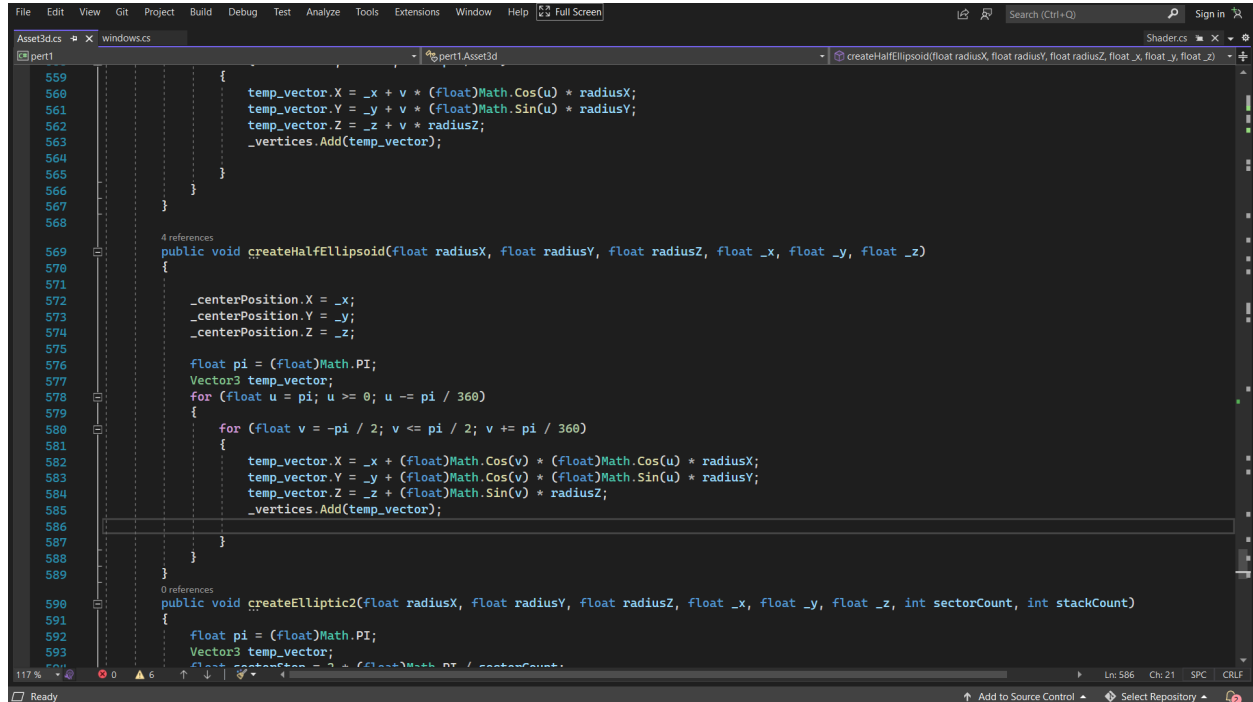
Merupakan fungsi yang digunakan untuk membuat bangun tabung. Dengan parameter yang diperlukan sebagai berikut top_radius, bot_radius, height, _x, _y, _z. Dengan top & bot radius untuk menentukan radius bagian atas dan bawah tabung. Height digunakan untuk menentukan tinggi tabung. X, y dan z digunakan untuk menentukan koordinat peletakan tabung.



```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Full Screen
Asset3d.cs x windows.cs Shader.cs x
pert1 Asset3d createCylinder2(float top_radius, float bot_radius, float height, float _x, float _y, float _z)
515
516
0 references
517 public void createCylinder2(float top_radius, float bot_radius, float height, float _x, float _y, float _z)
518 {
519     _centerPosition.X = _x;
520     _centerPosition.Y = _y;
521     _centerPosition.Z = _z;
522     float pi = (float)Math.PI;
523     Vector3 temp_vector;
524
525     for (float i = -pi / 2; i <= pi / 2; i += pi / 360)
526     {
527         for (float j = -pi; j <= pi; j += pi / 360)
528         {
529
530             temp_vector.Y = top_radius * (float)Math.Cos(i) * (float)Math.Sin(j) + _centerPosition.Y;
531             if (temp_vector.Y < _centerPosition.Y)
532             {
533                 temp_vector.Y = _centerPosition.Y - height * 0.5f;
534                 temp_vector.X = bot_radius * (float)Math.Cos(i) * (float)Math.Cos(j) + _centerPosition.X;
535                 temp_vector.Z = bot_radius * (float)Math.Sin(i) + _centerPosition.Z;
536             }
537             else
538             {
539                 temp_vector.X = top_radius * (float)Math.Cos(i) * (float)Math.Cos(j) + _centerPosition.X;
540                 temp_vector.Y = _centerPosition.Y + height * 0.5f;
541                 temp_vector.Z = top_radius * (float)Math.Sin(i) + _centerPosition.Z;
542             }
543             _vertices.Add(temp_vector);
544         }
545     }
546
547
0 references
548 public void createEllipticCone(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z)
549 {
550     _centerPosition.X = _x;
551
117% Lrn: 536 Ch: 22 SPC CRLF
Ready Add to Source Control Select Repository
```

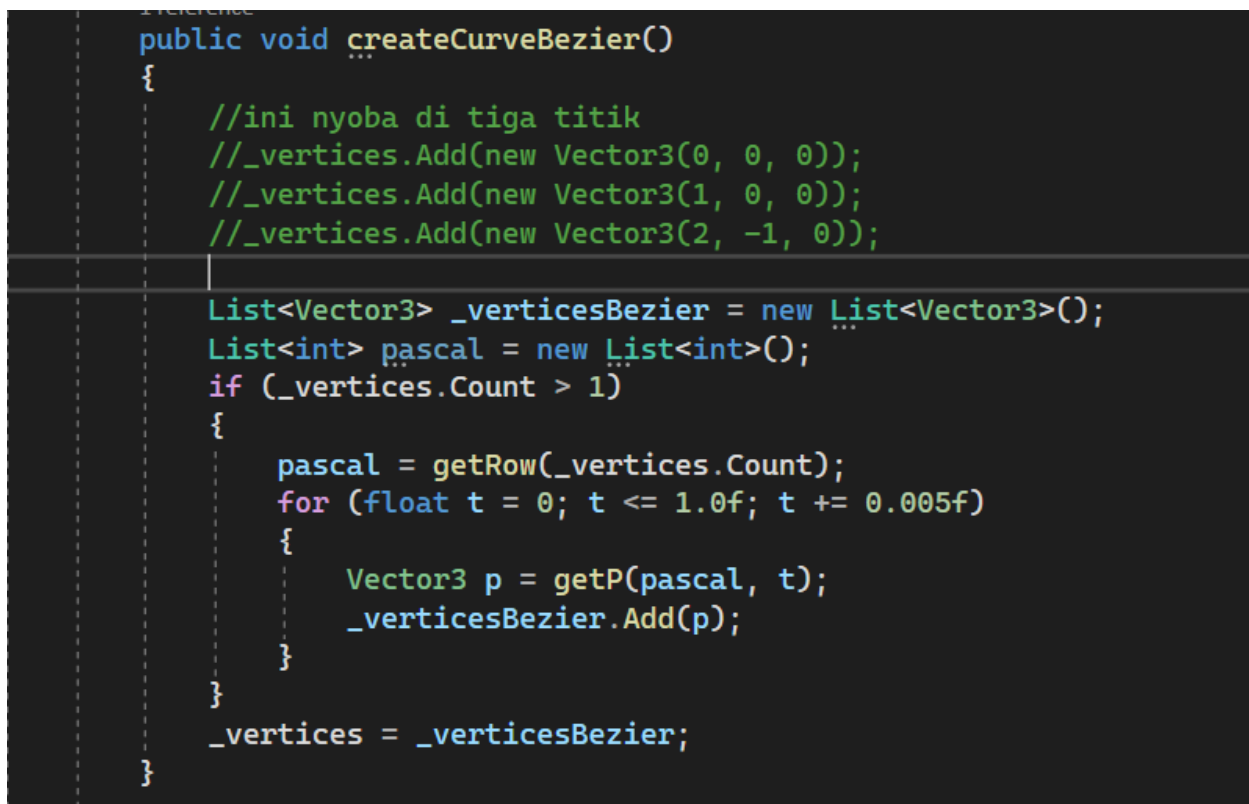
5. createHalfEllipsoid

Merupakan fungsi yang digunakan untuk membuat setengah elipsoid yang dimana menggunakan parameter radiusX, radiusY, radiusZ, _x, _y, _z. Dengan radiusX,y,z digunakan untuk menentukan radiusnya x,y, dan z sedangkan _x,_y, dan _z digunakan untuk menentukan koordinat half ellipsoid.



6. createCurveBezier

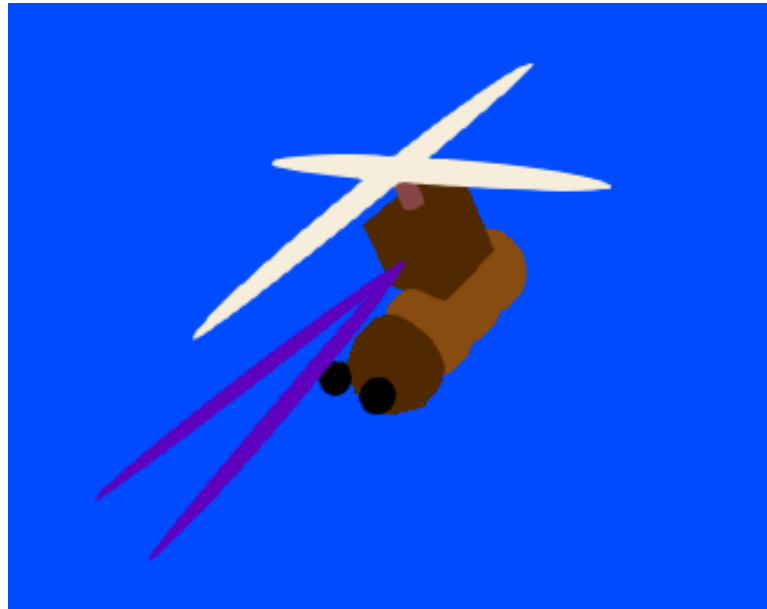
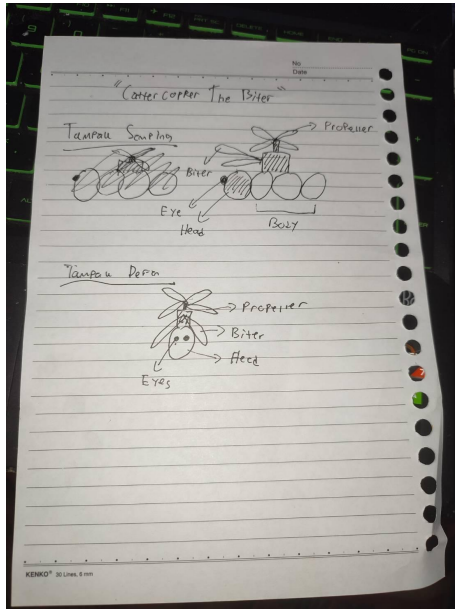
Fungsi ini digunakan untuk membuat model bezier.



B.Design Modeling

“Cattercopper The Biter”

By: Audrico

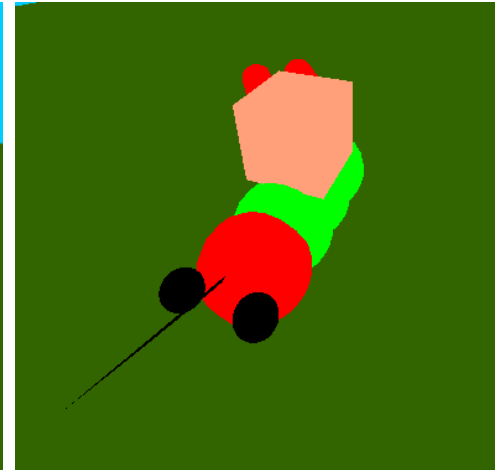
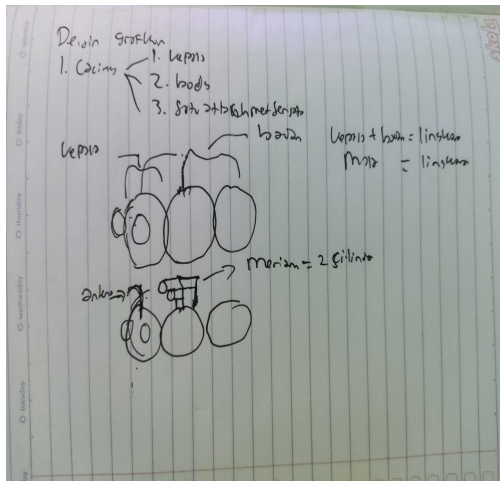


Object	Bagian Object	Penjelasan
Kepala	Kepala	Kepala menggunakan bentuk ellipsoid dengan fungsi createEllipsoid2 berparameter radiusX, y dan z = 0.3f dan menggunakan sectorcount dan stackCount = 10
	Mata	Kepala menggunakan bentuk ellipsoid dengan fungsi createEllipsoid berparameter radiusX, y dan z = 0.1f
Segmen Badan	Badan	Sama seperti kepala namun menggunakan warna yang beda
	Pangkuan capitan dan baling-baling	Menggunakan bentuk kubus dengan fungsi createboxvertices berparameter length = 0.5f
	Capitan	Menggunakan bentuk half ellipsoid dengan fungsi createHalfEllipsoid berparameter radiusX = 0.03f, radiusY= 0.1f dan radiusZ = 1.0f

	Tangkai baling-baling	Menggunakan bentuk balok dengan fungsi createRectangularVertices berparameter length = 0.1f
	Baling-baling	Menggunakan bentuk half elipsoid dengan fungsi createHalfEllipsoid berparameter radiusX = 0.1f, radiusY= 0.1f dan radiusZ = 1.5f

“Missile worm ”

By: Ramadhan Daffa

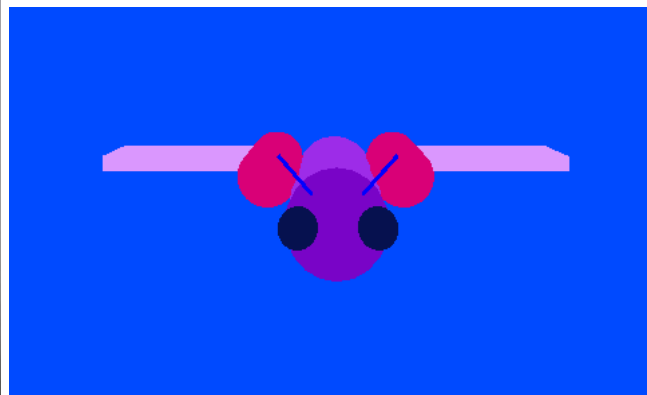
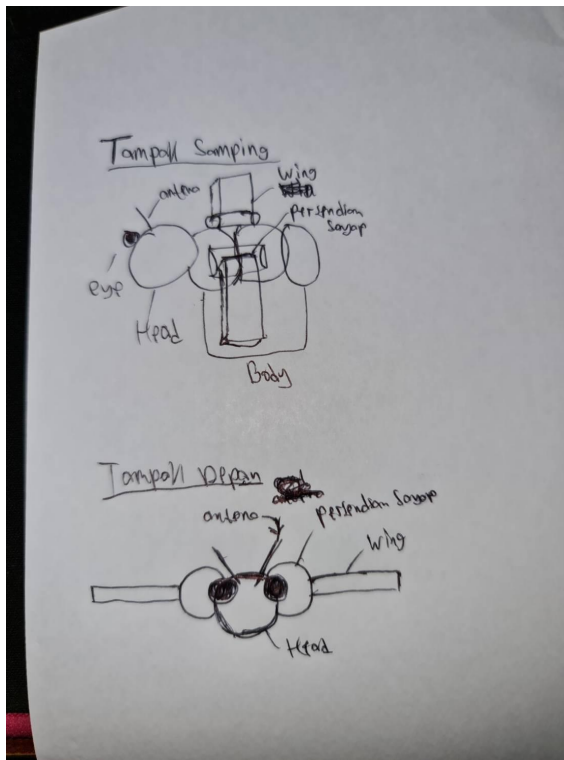


Object	Bagian Object	Penjelasan
Kepala	kepala	Dibangun menggunakan ellipsoid2 dengan parameter $radx,y,z=0.3$, koordinat $x = -0.9$, $y = 0$ $z = 0$ stack =10 count =10
	Mata	Dibangun dengan createellipsoid parameter $radx,y,z=0.1$, koordinat $x = -1.2$, $y = 0.1$, $z = 0.2$
	mata2	Dibangun dengan createellipsoid parameter $radx,y,z=0.1$, koordinat $x = -1.2$, $y = 0.1$, $z = -0.2$
	antena	Menggunakan fungsi create bezier dengan 3 vertex $(-0.75f, 0.05f, 0f)$, $(-0.75f, 0.1f, 0f)$, $(-0.6f, 0.15f, 0f)$
badan	badan1	dibangun menggunakan ellipsoid2 dengan parameter $radx,y,z=0.3$, koordinat $x = -0.6$, $y = 0$ $z = 0$ stack =10 count =10
	badan2	dibangun menggunakan ellipsoid2 dengan parameter $radx,y,z=0.3$, koordinat $x = -0.3$, $y = 0$ $z = 0$ stack =10 count =10
	badan3	dibangun menggunakan ellipsoid2 dengan parameter $radx,y,z=0.3$, koordinat $x = 0$, $y = 0$ $z = 0$ stack =10 count =10

Senjata	Kotak	Dibangun dengan createboxvertice dengan parameter Koor x = -0.3 koor y = 0.5 koor z = 0.0 lenght = 0.5
	silinder1	Dibangun dengan createsilinder dengan parameter Top dan bot radius = 0.1 height = 0.5 dan koor xyz = -0.7f, 0.6f, 0.15f
	silinder 2	Dibangun dengan createsilinder dengan parameter Top dan bot radius = 0.1 height = 0.5 dan koor xyz = -0.7f, 0.6f, -0.15f

“Plane Worm”

By: Fernando



Object	Bagian Object	Penjelasan
Kepala	Kepala	Kepala menggunakan bentuk ellipsoid dengan fungsi createEllipsoid2 berparameter radiusX, y dan z = 0.3f dan menggunakan sectorcount dan stackCount = 10
	Mata	Kepala menggunakan bentuk ellipsoid dengan fungsi createEllipsoid berparameter radiusX, y dan z = 0.1f
Segmen Badan	Badan	Dibangun menggunakan ellipsoid2 dengan parameter radx,y,z=0.3,koordinat x = -0.6 , y z = 0 stack =10 count =10
	sayap	Menggunakan bentuk kubus dengan fungsi createboxvertices berparameter length = 0.3f
	Engsel sayap	Menggunakan bentuk ellipsoid dengan fungsi createCylinder2 berparameter radiusX = -0.3f, radiusY= 0.15f dan radiusZ = 0.45f
	anthena	Menggunakan bentuk tabung dengan fungsi createCylinder2 berparameter (Top_radius = 0.01f,Bot_radius = 0.01f,height = 0.8f, X= -1.0f, Y = 0.15f, Z = 0.1f)

C.ANIMASI

Model	ID Animasi	Deskripsi
Cattercopper The Biter	rotateheli	Merotasi body objek terhadap sumbunya sendiri secara rekursif hingga $\text{time_render1} > 18$. Lalu mengubah variable $\text{done} = \text{True}$
Cattercopper The Biter	animasiBaling	<p>Mengubah variable $\text{done} = \text{False}$ lalu melakukan pengecekan terlebih dahulu. Jika $\text{time_render1} \leq 30f$ maka melakukan rotasi baling-baling terhadap sumbu tiang baling-baling, melakukan translasi $y = 0.1f$ lalu menambahkan $\text{time_render1} += 1$.</p> <p>Pada saat $\text{time_render1} > 30f$ akan melakukan rotasi baling-baling seperti pertama lalu melakukan rotasi lagi terhadap sumbu pohon paling kiri</p>
Missile worm	Jalan(if pertama)	<p>Worm melakukan gerakan maju dan mundur untuk melakukannya</p> <p>Dibuat pengecekan dari koordinat z dari worm dengan pengecekan jika kurang dari -2 atau lebih dari 2 maka $\text{time_translasi} = 1$ pada kondisi kurang dari 2, $\text{time_translasi} = 1$ pada jika lebih dari -2 lalu di luar if worm akan di translasi z dan dikali time_translasi pengecekan di animasi ini masih bisa dirubah sesuai kebutuhan</p>
Missile Worm	Jalan(if kedua)	<p>Di saat worm melakukan translasi di if kedua akan membuat rotasi silinder senjata nya untuk berputar pada sumbu dari kubus dengan jalur silinder masing masing. Pengecekan posisi</p> <p>Jika posisi worm kurang dari -0.5 maka perputaran akan mengikuti jarum jam sedangkan jika kurang dari 0.5 maka akan sebaliknya</p>
Plane Worm	rotateSayap	<p>Pada saat objek pertama kali dijalankan, variable $\text{done} = \text{false}$, yang berarti akan melakukan rotasi berulang terhadap sumbunya sendiri dan melakukan penambahan variable $\text{time_render3} += 1$ hingga $\text{time_render3} > 12f$ akan menghentikan proses rekursi dan mengubah variable done menjadi True</p>
Plane Worm	rotatepesawat	<p>Pada saat variable $\text{done} = \text{true}$, fungsi akan melakukan rotasi terhadap sumbu pohon paling kiri</p>