

TANGGAL PERCOBAAN : 07 Desember 2022
TANGGAL PENGUMPULAN : 24 Desember 2022

**LAPORAN AKHIR PROYEK KELOMPOK PEMROGRAMAN
KOMPUTER**

TRANSVERSE WAVES USING CIRCLE



*Mencerdaskan &
Memartabatkan Bangsa*

KELOMPOK : 8 – Delapan

NAMA : 1. Dhinar hidayath (1306621054)
2. Khansa Farras Callista (1306621067)
3. Salma Mardhiyah (1306621061)

DOSEN PENGAMPU : Drs. Andreas Handjoko Permana, M.Si

ASISTEN LABORATORIUM :

Ivan Hary Syahbana - 1302620050
Made Ocha Wiyadnyana - 1302620051
Tannie Wiyuna - 1302620044

Laporan Awal	Laporan Akhir	Total

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Jakarta

2022

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Konsep gelombang dalam fisika bisa dikatakan cukup membuat siswa maupun mahasiswa kebingungan. Tidak jarang, mahasiswa kebingungan ketika ditanya mengenai jenis gelombang oleh dosen. Banyak mahasiswa yang masih kesulitan dalam membedakan jenis gelombang karena kurangnya pemahaman konsep gelombang.

Dalam menerapkan ilmu fisika, baik penyelesaian soal atau pun pengamatan objek yang berhubungan dengan fisika tentunya dipengaruhi oleh penguasaan dan pemahaman konsep fisika. Pada kenyataannya masih banyak mahasiswa yang belum menguasai konsep fisika seutuhnya, fenomena ini dikenal dengan *misconception*. Menurut Suparno Miskonsepsi adalah suatu konsep pemahaman yang tidak sesuai dengan konsep yang lazim diakui oleh para ahli. Miskonsepsi dapat dialami oleh mahasiswa, karena miskonsepsi dapat bersumber dari pendidikan atau pengalaman yang didapatkan sebelumnya.

Gelombang merupakan salah satu objek yang dikaji dalam ilmu Fisika. Secara Fisika, gelombang adalah getaran yang merambat. Gelombang merambat menggunakan media atau pun tidak. Berdasarkan arah rambat dan getarannya, gelombang terbagi menjadi dua jenis yaitu gelombang transversal dan gelombang longitudinal.

Gerak melingkar adalah pergerakan dari suatu objek yang membentuk lintasan berupa lingkaran dengan titik tetap atau sumbu di tengahnya. Gerak melingkar terbagi menjadi dua yaitu Gerak Melingkar Beraturan yang dikenal dengan GMB dan Gerak Melingkar Berubah Beraturan yang dikenal dengan GMBB. GMB merupakan gerak melingkar yang tidak mengalami perubahan kecepatan (percepatan) sedangkan GMBB merupakan gerak melingkar yang mengalami perubahan kecepatan.

Dalam rangka memenuhi penugasan Pemrograman Komputer akan dibuat simulasi gelombang menggunakan python dengan tampilan yang menarik dan interaktif. Simulasi gelombang yang akan dibuat merupakan gabungan dari materi gerak melingkar dan gelombang. Gelombang yang akan digunakan pada simulasi adalah gelombang transversal yang terbentuk dari pergerakan gerak melingkar suatu

lingkaran yang mengalami percepatan. Tampilan simulasi akan dibuat menarik dengan tambahan gambar dan suara.

1.2 PERUMUSAN MASALAH

1. Bagaimana hubungan antara jari - jari lingkaran dan gelombang?
2. Gelombang apa yang terbentuk dari gerakan lingkaran?
3. Bagaimana proses penggambaran simulasi gelombang dapat terbentuk dengan menggunakan program python?
4. Apa yang terjadi jika kecepatan pergerakan lingkaran dipercepat atau diperlambat?

1.3 TUJUAN

1. Memenuhi penugasan mata kuliah Pemrograman Komputer
2. Menambah pemahaman mahasiswa dalam mengidentifikasi jenis gelombang
3. Menambah pemahaman mahasiswa mengenai amplitudo gelombang
4. Menemukan hubungan pergerakan lingkaran dan gelombang
5. Menggabungkan konsep Gerak Melingkar Berubah Beraturan dan konsep Gelombang

BAB II

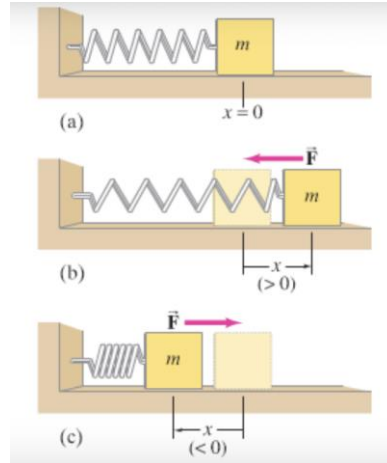
KAJIAN PUSTAKA

2.1 DASAR TEORI

Gelombang merupakan sebuah getaran yang merambat melalui media tertentu. Setiap benda yang berjalan dicirikan mempunyai kecepatan. Kecepatan gelombang bergantung pada sifat medium, dimana ia merambat. Pada saat gelombang merambat terdapat energi yang dipindahkan dari suatu tempat ke tempat yang lain **(King, 2009)**. Bentuk gelombang dibagi menjadi dua yaitu, gelombang longitudinal dan transversal. Gelombang longitudinal merupakan gelombang dengan arah gangguan sejajar dengan arah penjalarannya. Contoh dari gelombang longitudinal yaitu gelombang bunyi **(Handika & Fatahillah, 2019)**. Gelombang transversal merupakan gelombang yang arahnya rambatnya tegak lurus dengan arah getarannya. Getaran dapat dianggap sebagai gerakan berulang dari suatu objek di sekitar suatu posisi kesetimbangan atau equilibrium. Gelombang transversal dan longitudinal dibedakan berdasarkan pada arah rambat dan arah getarnya. Arah getar dan arah rambat gelombang transversal saling tegak lurus dan arah getar dan arah rambat gelombang longitudinal paralel **(Li, Liu, Chen, & Wan, 2020)**.

Ketika sebuah benda bergetar atau berosilasi bolak balik, maka gerakan tersebut dinamakan gerakan periodic. Dalam gambar, massa diabaikan dan pegas dipasang mendatar sehingga benda bermassa m meluncur tanpa gesekan pada permukaan mendatar. setiap pegas memiliki panjang alami di mana ia tidak memberikan gaya pada massa m . posisi massa pada titik ini disebut posisi equilibrium. Ketika benda ditarik maka akan pegas memberikan gaya pada massa dengan mengembalikan arah massa ke posisi equilibrium. gaya ini disebut gaya pemulih.

$$F = -kx$$



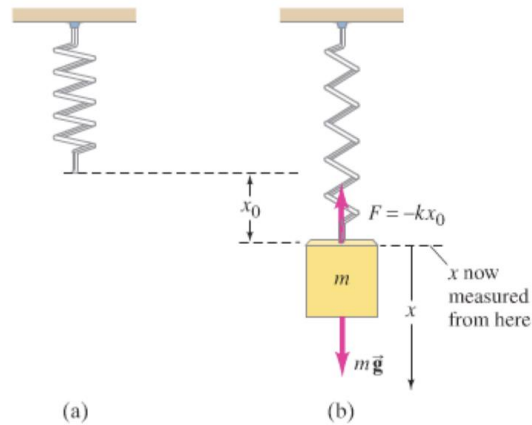
Gambar 1. Getaran massa pada pegas

Jarak x massa dari titik equilibrium pada setiap titik disebut perpindahan. perpindahan maksimum atau jarak terbesar dari titik equilibrium disebut amplitudo (A). Periode didefinisikan sebagai waktu yang diperlukan untuk menyelesaikan siklus per detik sedangkan frekuensi didefinisikan sebagai berapa siklus yang dapat dilakukan per waktunya atau per detiknya.

$$f = \frac{n}{t} \quad T = \frac{t}{n}$$

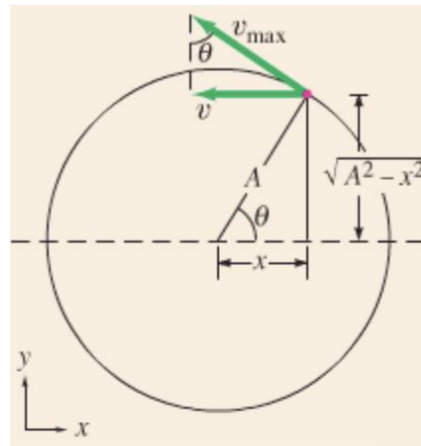
$$f = \frac{1}{T} \quad T = \frac{1}{f}$$

Pegas dan massa yang diletakkan secara vertikal pada dasarnya sama dengan horizontal. Namun karena gravitasi, panjang pegas vertikal dengan massa m akan lebih panjang pada equilibrium daripada pegas yang horizontal, seperti yang ditunjukkan pada gambar 2. Pegas akan pada equilibrium ketika $\Sigma F = 0 = mg - kx_0$ sehingga pegas akan tertarik secara maksimum ketika $x_0 = \frac{mg}{k}$ supaya equilibrium.



Gambar 2. Pegas dan massa secara vertikal

Gerak melingkar dapat digunakan untuk menggambarkan gerak harmonik sederhana. Gerakan memutar ini mendefinisikan gerak harmonik sederhana dari amplitudo (A) dan kecepatan sudut (ω) pada garis lingkaran (**French, 2017**). Dengan meletakkan sebuah benda bermassa m yang berputar berlawanan jarum jam dengan jari-jari A dengan kecepatan konstan v_{max} seperti pada gambar, maka akan didapatkan



Gambar 3. Gerak melingkar

Dari gambar terlihat bahwa $\cos \theta = \frac{x}{A}$, jadi proyeksi posisi benda pada sumbu x .

$$x = A \cos \theta$$

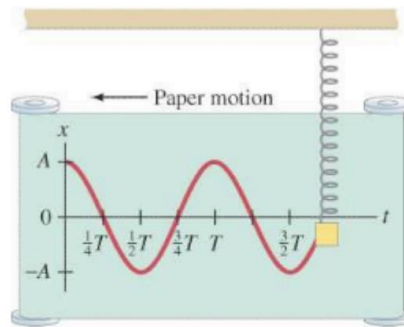
Karena massa berotasi dengan kecepatan sudut, maka $\theta = \omega t$ dimana θ radian.

$$x = A \cos \omega t$$

$$\omega = 2\pi f$$

$$x = A \cos 2\pi f t$$

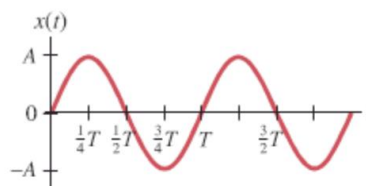
$$x = A \cos \frac{2\pi t}{T}$$



Gambar 4. Hubungan pegas dengan gelombang

Komponen x dari gerak benda yang berotasi berkorespondensi persis dengan gerak osilator harmonik sederhana. Dengan demikian persamaan $x = A \cos \frac{2\pi t}{T}$ memberikan posisi suatu benda yang mengalami gerak harmonik. Persamaan tersebut menunjukkan bahwa benda berosilasi dari $x = A$ dimana $t = 0$. Apabila ketika $t = 0$ objek berada pada $x = 0$ maka persamaan akan menjadi,

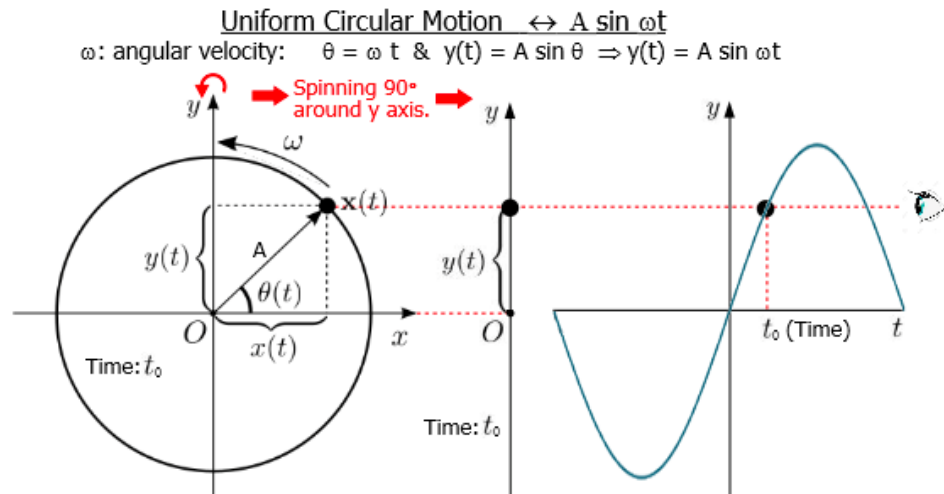
$$x = A \sin \omega t = A \sin \left(\frac{2\pi t}{T} \right)$$



Gambar 5. Gelombang sinusodial

Gambar gelombang tersebut memiliki bentuk yang mirip dengan gelombang \cos . Yang membedakan hanya letak titik x awal atau ketika $t = 0$, dimana x terletak pada A untuk gelombang \cos dan x terletak pada 0 ketika gelombang \sin (**Giancoli 2004**). Kedua gelombang tersebut merupakan gelombang sinusoidal yang juga termasuk ke dalam gelombang transversal.

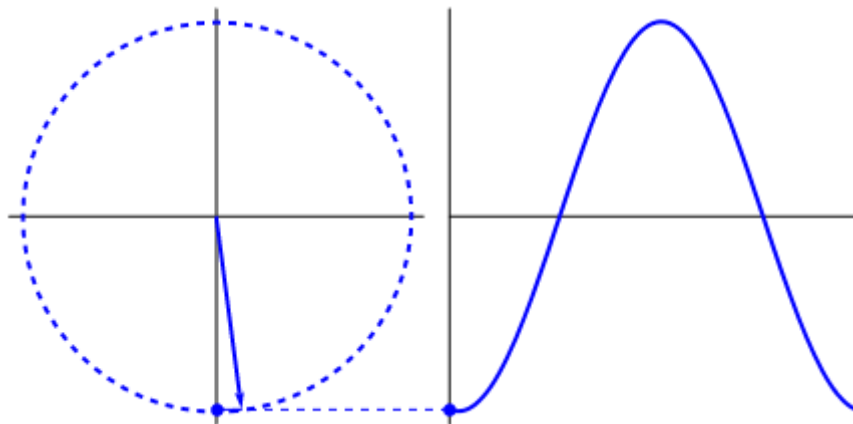
Selain dari gerak pegas, gerak melingkar pada lingkaran juga dapat berkorespondensi dengan gerak osilator harmonik sederhana. Dengan meletakkan sebuah benda yang bergerak berlawanan jarum jam dengan jari-jari A dengan kecepatan sudut ω konstan seperti pada gambar. Apabila benda tersebut ditarik garis ke sumbu y maka akan tercipta gerak harmonik. Gerakan ini mirip dengan gerak pegas.



Gambar 6. Hubungan gerak melingkar dengan gelombang

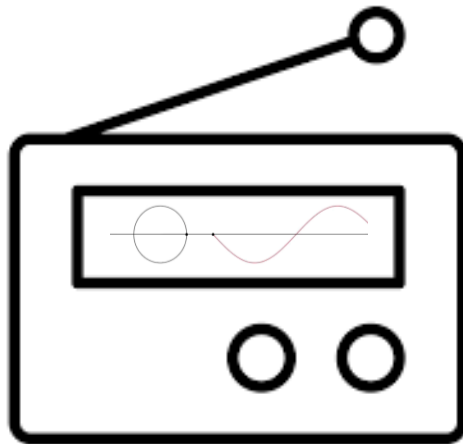
2.2 RENCANA SIMULASI

Ketika kita menarik garis dari pusat ke ujung lingkaran maka tercipta garis jari-jari lingkaran atau r . Ketika garis tersebut berotasi dan kita menarik garis ke titik sumbu y maka titik tersebut akan membentuk sebuah gerak gelombang.



Kecepatan rotasi tersebut maka akan berpengaruh kepada kerapatan gelombang. Sehingga apabila kecepatan rotasi garis r dipercepat maka gelombang akan semakin rapat dan ketika kecepatan rotasi garis r diperlambat maka gelombang akan semakin renggang. Panjang dari garis r tersebut juga berpengaruh terhadap amplitudo gelombang tersebut. Sehingga apabila panjang garis r diperpanjang maka amplitudo gelombang semakin besar dan apabila panjang garis r diperkecil atau diperpendek maka amplitudo gelombang akan semakin kecil.

Agar lebih menarik, kami menyambungkan gelombang tersebut dengan musik sehingga tampilannya akan seperti radio dimana volume dari musik akan mempengaruhi kecepatan rotasi garis r . Sehingga ketika suara dari musik tersebut diperbesar maka gelombang akan semakin rapat dan ketika suara dari musik tersebut diperkecil maka gelombang akan semakin renggang.



BAB III

LANGKAH-LANGKAH PEMROGRAMAN

3.1 PROBLEM STATEMENT

Membuat simulasi gelombang transversal yang terbentuk dari pergerakan sebuah lingkaran.

3.2 MATHEMATICAL EQUATION

- 1) $x += (\text{radius}/i) * \text{math.cos}(\text{angle}*i)$
- 2) $y += (\text{radius}/i) * \text{math.sin}(\text{angle}*i)$
- 3) $\text{radius} += \text{kenaikan_radius}$
- 4) $\text{radius} -= \text{kenaikan_radius}$
- 5) $\text{kecepatan_sudut} += \text{kenaikan_sudut}$
- 6) $\text{kecepatan_sudut} -= \text{kenaikan_sudut}$

3.3 ALGORITMA

1. Mulai
2. Memanggil import pygame.gfxdraw, pygame.constants, pygame.locals, sys, pygame.locals
3. Memanggil from pygame import mixer
4. Memanggil from ctypes import cast, POINTER
5. Memanggil from ctypes import CLSCTX_ALL
6. Memanggil from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume
7. Menginisiasikan pygame.init()
8. Menginisiasikan warna yang dibutuhkan
9. Mendefinisikan ukuran layar (size) = 1500, 900
10. Mendefinisikan surface = pygame.Surface(size)
11. Mendefinisikan screen = pygame.display.set_mode(size)
12. Menampilkan judul pada jendela layar sebelah pojok kiri dengan kalimat 'Tranverse Wave Using Circle'
13. Mendefinisikan FPS = 100 dan clock = pygame.time.Clock()
14. Menginisiasikan pause = False, radius = 35, kenaikan_radius = 5, angle = 0, dan n = 0

15. Menginisiasikan kecepatan sudut lingkaran = 0,005 dengan kenaikan_sudut sebesar 0,0049, center = 50, 485
16. Membuat list wave = []
17. Mendefinisikan speaker dan mengatur volume
18. Melakukan proses fungsi mixer.music.load("sound1.mp3) untuk memasukan suara
19. Mendefinisikan batas gelombang = 403
20. Mendefinisikan garis_penghubung lingkaran dengan grafik = center[0] + 300
21. Melakukan pengondisian looping while True
 - 21.1. Mendefinisikan bg untuk background tampilan layar dengan fungsi bg = pygame.image.load("bg.jpg").convert()
 - 21.2. Menampilkan background dan surface
 - 21.3. Memperbarui display pada window dengan perintah pygame.display.flip()
 - 21.4. Mengatur kecepatan tampilan gambar dengan perintah clock.tick(FPS)
 - 21.5. Melakukan pengondisian looping for event in pygame.event.get():
 - 21.5.1. Melakukan pengondisian if event.type == pygame.QUIT
 - 22.5.1.1 Melakukan proses pygame.quit()
 - 22.5.1.2 volume.SetMasterVolumeLevel(-15.0, None)
 - 22.5.1.3 Melakukan proses sys.exit()
 - 21.5.2. Melakukan pengondisian if pygame.key.get_pressed()[pygame.K_p]
 - 22.5.2.1 Suara diputar
 - 22.5.2.2 n =1
 - 21.5.3. Melakukan pengondisian if pygame.key.get_pressed()[pygame.K_SPACE]
 - 22.5.3.1 pause = True
 - 22.5.3.2 Suara berhenti
 - 21.5.4. Melakukan pengondisian if pygame.key.get_pressed()[pygame.K_UP]
 - 22.5.4.1 radius += kenaikan_radius
 - 22.5.4.2 Melakukan pengondisian if radius >70
 - 22.5.4.2.1 radius = 70

- 21.5.5. Melakukan pengondisian if `pygame.key.get_pressed()[pygame.K_DOWN]`
 - 22.5.5.1 `radius -= kenaikan_radius`
 - 22.5.5.2 Melakukan pengondisian if `radius < 35`
 - 22.5.5.2.1 `radius = 35`
- 21.5.6. Melakukan pengondisian if `pygame.key.get_pressed()[pygame.K_RIGHT]`
 - 22.5.6.1 `kecepatan_sudut += kenaikan_sudut`
 - 22.5.6.2 Melakukan pengondisian if `kecepatan_sudut > 0.25`
 - 22.5.6.2.1 `kecepatan_sudut = 0.25`
 - 22.5.6.2.2 Melakukan proses menaikkan volume
- 21.5.7. Melakukan pengondisian if `pygame.key.get_pressed()[pygame.K_LEFT]`
 - 22.5.7.1 `kecepatan_sudut -= kenaikan_sudut`
 - 22.5.7.2 Melakukan pengondisian if `kecepatan_sudut < 0.005`
 - 22.5.7.2.1 `kecepatan_sudut = 0.005`
 - 22.5.7.2.1 Melakukan proses menurunkan volume
- 21.6. Melakukan pengondisian looping while pause
 - 21.13.1. Melakukan pengondisian looping for event in `pygame.event.get()`
 - 21.6.1.1. Melakukan pengondisian if `event.type == pygame.QUIT`
 - 22.6.1.1.1 Melakukan proses `pygame.quit()`
 - 22.6.1.1.2 Mengatur volume akhir perangkat
 - 22.6.1.1.3 Melakukan proses `sys.exit()`
 - 21.6.1.2. Melakukan pengondisian elif `pygame.key.get_pressed()[pygame.K_SPACE]`
 - 22.6.1.2.1 Menginisiasikan `pause = False`
 - 22.6.1.2.2 Menjalankan kembali program
- 21.7. Mengatur jenis dan ukuran font
- 21.8. Menginisiasikan teks `kecepatan_sudut = font.render(f'{round(kecepatan_sudut, 10)}', True, toska)`

- 21.9. Menginisiasikan `amplitudo = font.render(f"{{round(radius,10)}}", True, toska)`
- 21.10. Memberikan surface berwarna hitam (0,0,0)
- 21.11. Menampilkan teks `kecepatan_sudut` dan `amplitudo` pada surface
- 21.12. Menginisiasikan `x, y = center`
- 21.13. Melakukan pengondisian looping `for i in range(1, 2*n + 1, 2)`
 - 21.13.1. Menginisiasikan `prevx, prevy = x, y`
 - 21.13.2. Menghitung `x += (radius / i) * math.cos(angle * i)`
 - 21.13.3. Menghitung `y += (radius / i) * math.sin(angle * i)`
 - 21.13.4. Membuat garis batas lingkaran dengan fungsi `pygame.gfxdraw.aacircle(surface, (prevx+600), (prevy), radius, yellow)`
 - 21.13.5. Membuat garis batas lingkaran dengan fungsi `pygame.gfxdraw.aacircle(surface, (prevx+600), (prevy), radius+1, yellow)`
 - 21.13.6. Membuat garis dalam lingkaran atau jari-jari dengan fungsi `pygame.draw.aaline(surface, yellow, (prevx+600, prevy), (x+600, y))`
 - 21.13.7. Membuat garis dalam lingkaran atau jari-jari dengan fungsi `pygame.draw.aaline(surface, yellow, (prevx+601, prevy), (x+601, y))`
- 21.14. Menghitung `angle += kecepatan_sudut`
- 21.15. Melakukan proses `wave.insert(0,y)`
- 21.16. Menggambar garis penghubung dengan fungsi `pygame.draw.aaline(surface, toska, (x+600, y), (garis_penghubung+420, wave[0]))`
- 21.17. Menggambar bulatan dalam lingkaran dengan fungsi `pygame.draw.circle(surface, yellow, [x+600, y], 5)`
- 21.18. Melakukan pengondisian `if len(wave)>batas_gelombang` maka `wave.pop()`
- 21.19. Menginisiasikan `titik_wave = [(garis_penghubung + j + 420, w) for j, w in enumerate(wave)]`
- 21.20. Melakukan pengondisian `if len(titik_wave) > 1`
 - 21.20.1 Menggambar gelombang dengan fungsi `pygame.draw.aalines(surface, toska, False, titik_wave)`

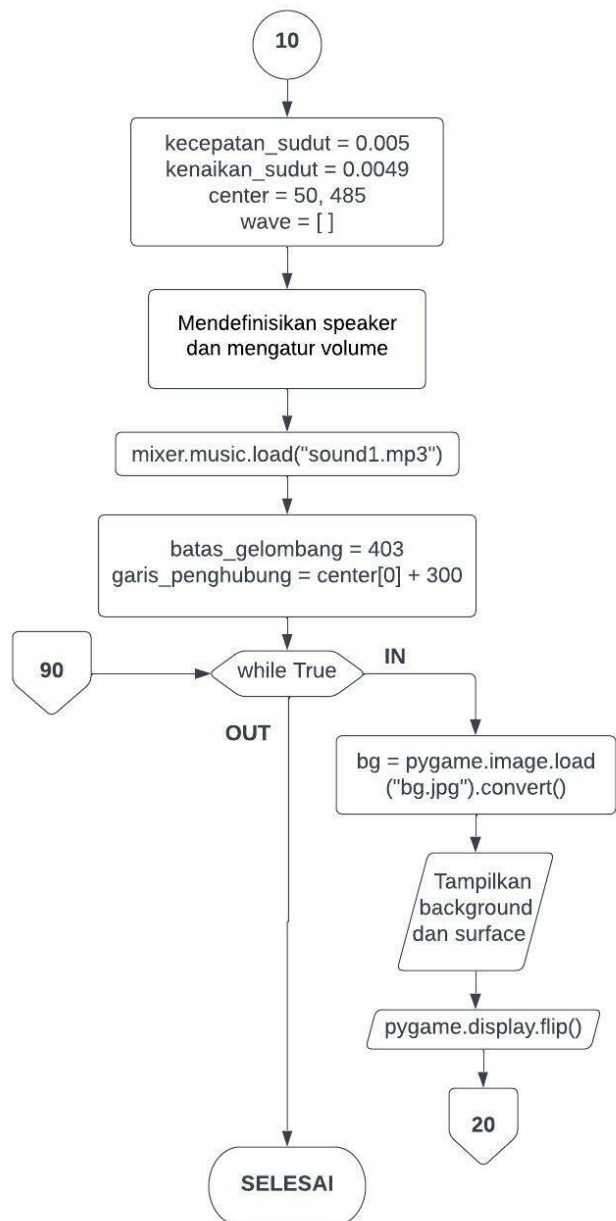
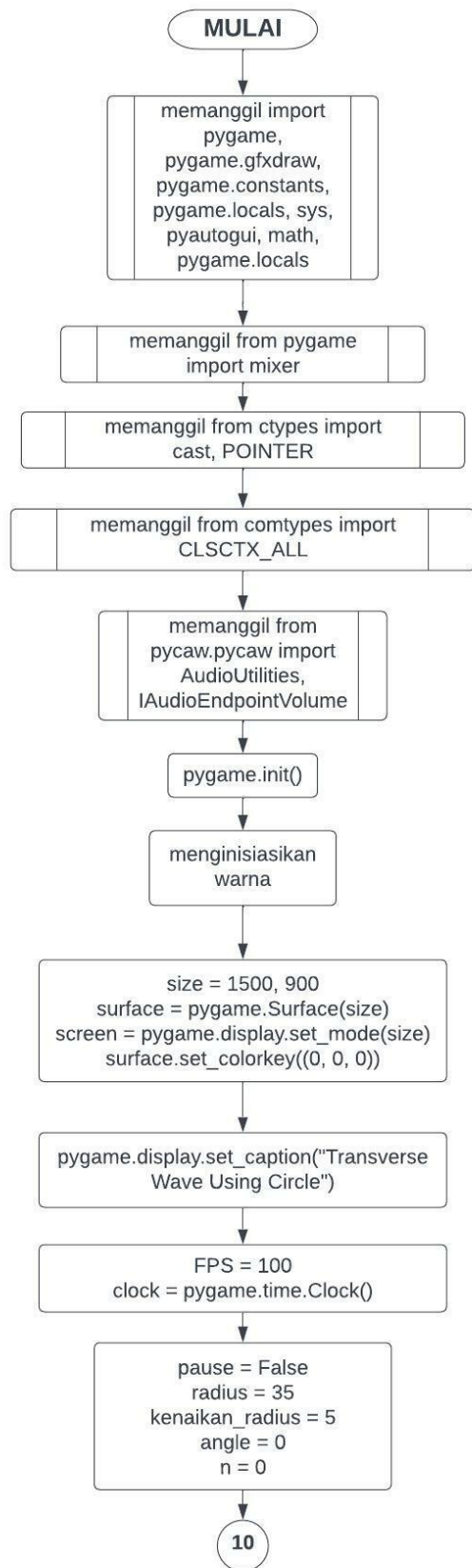
21.20.1 Menggambar gelombang dengan fungsi `pygame.draw.aalines`
(surface, toscs , False, [(garis_penghubung + j + 420, w + 1.2)
for j, w in enumerate(wave)])

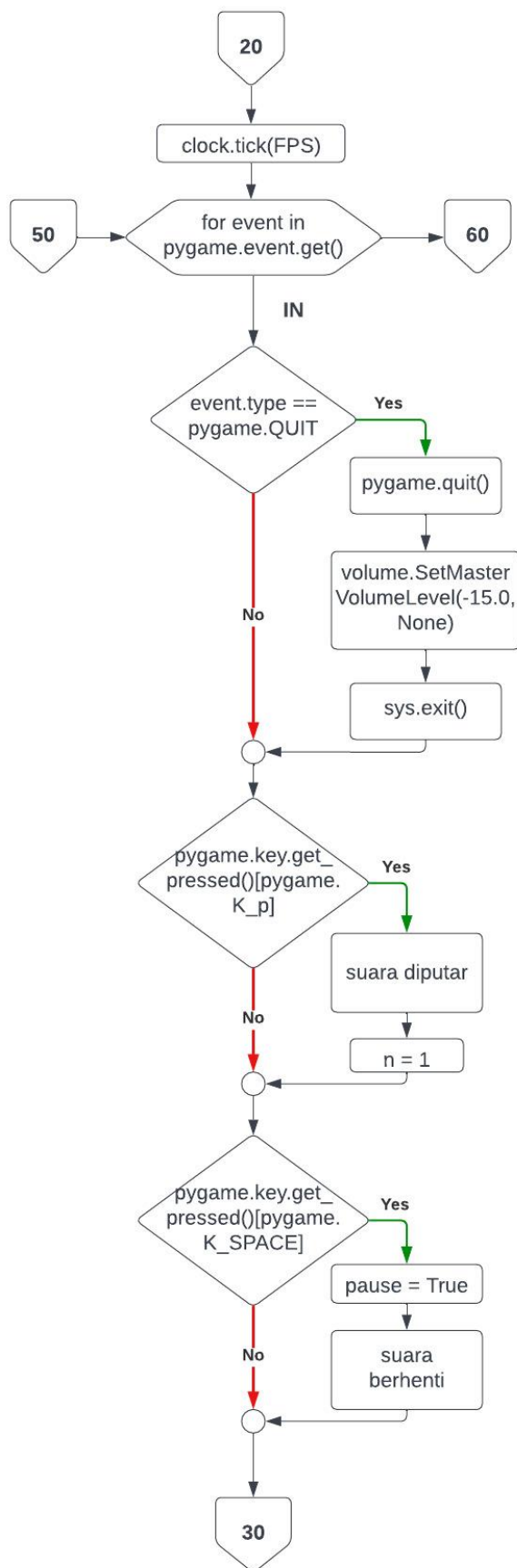
21.20.1 Menggambar gelombang dengan fungsi `pygame.draw.aalines`
(surface, toscs , False, [(garis_penghubung + j + 420, w + 1.8)
for j, w in enumerate(wave)]))

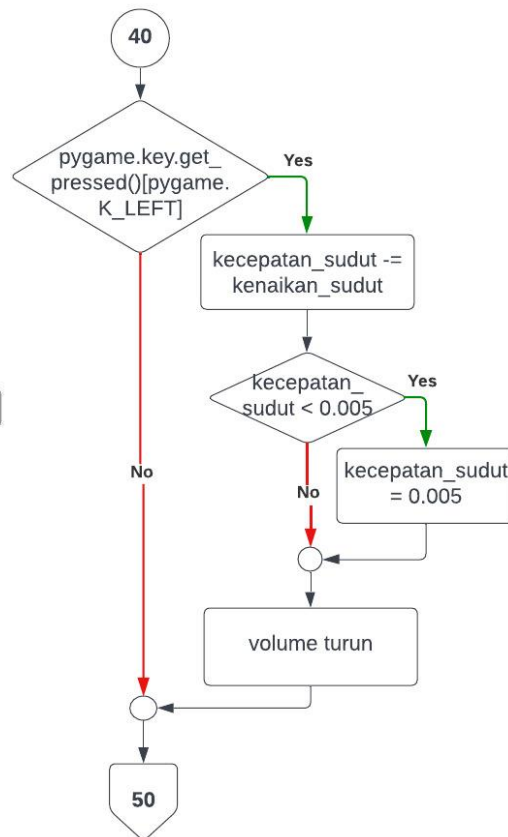
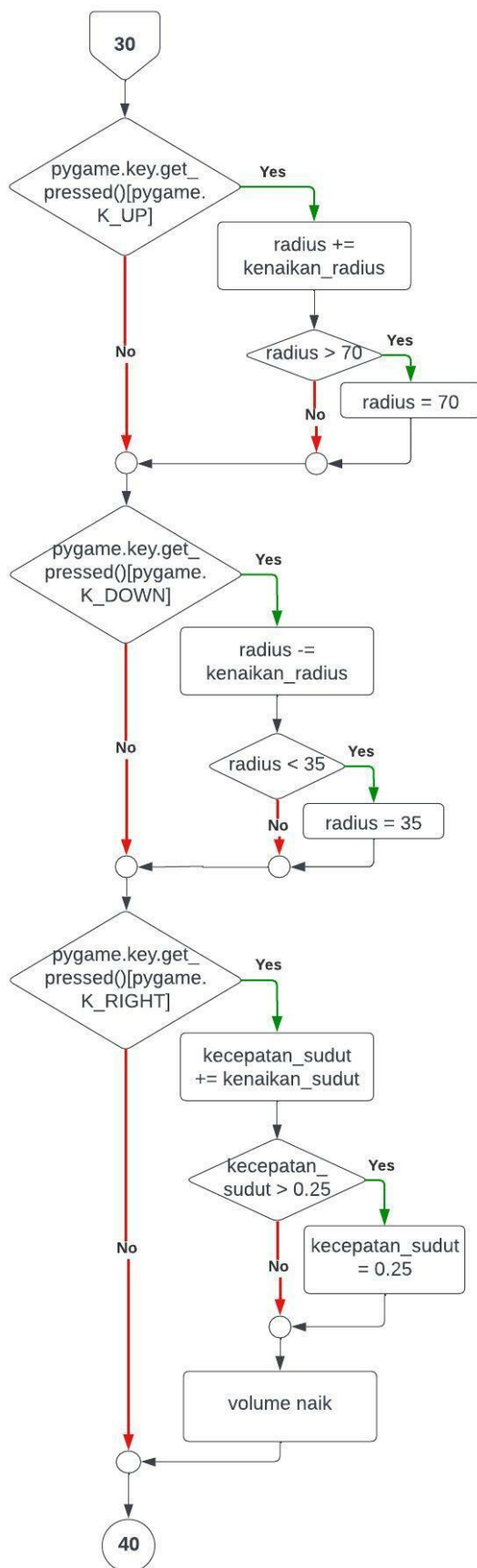
22. Selesai

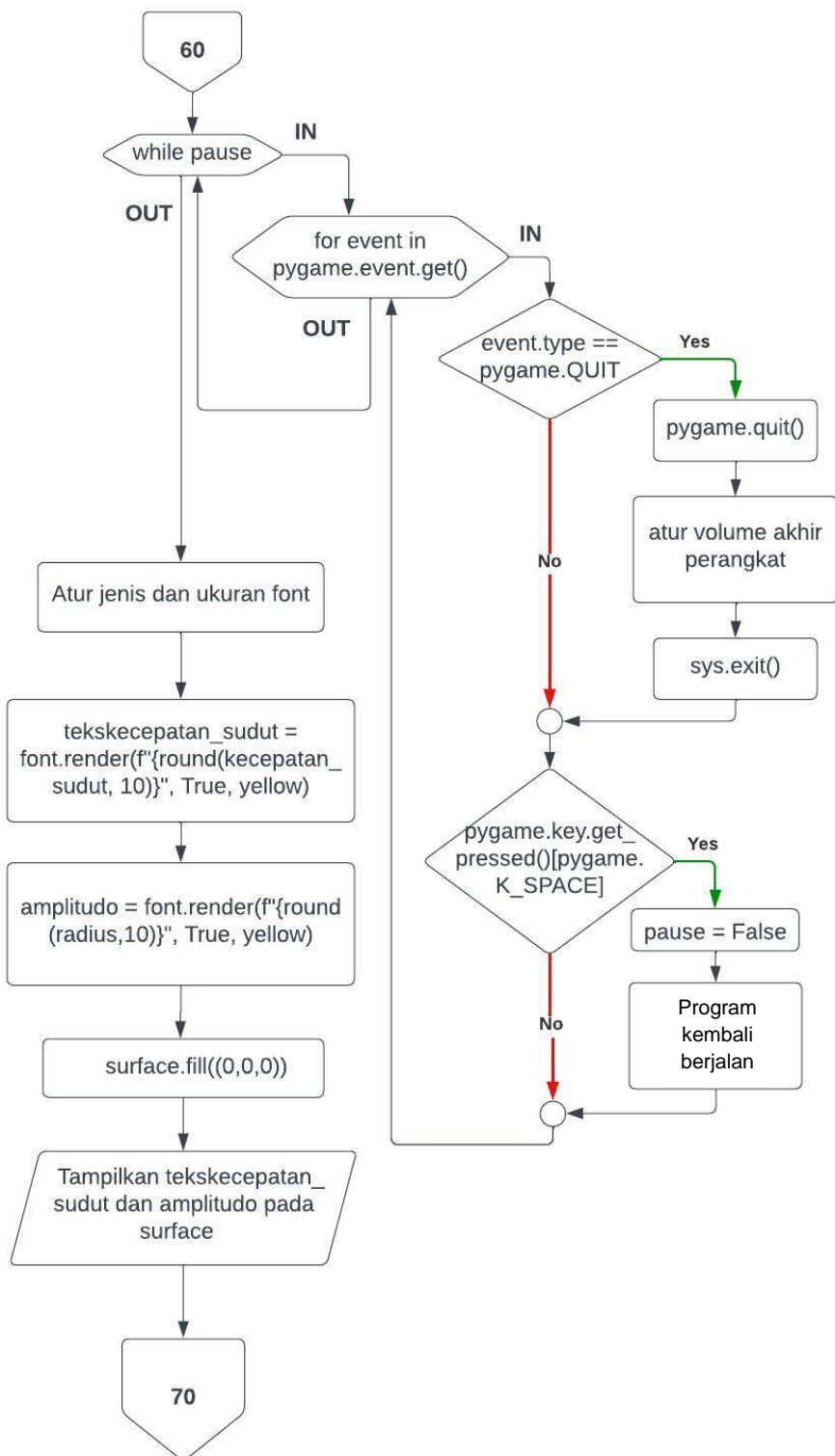
3.4 FLOWCHART

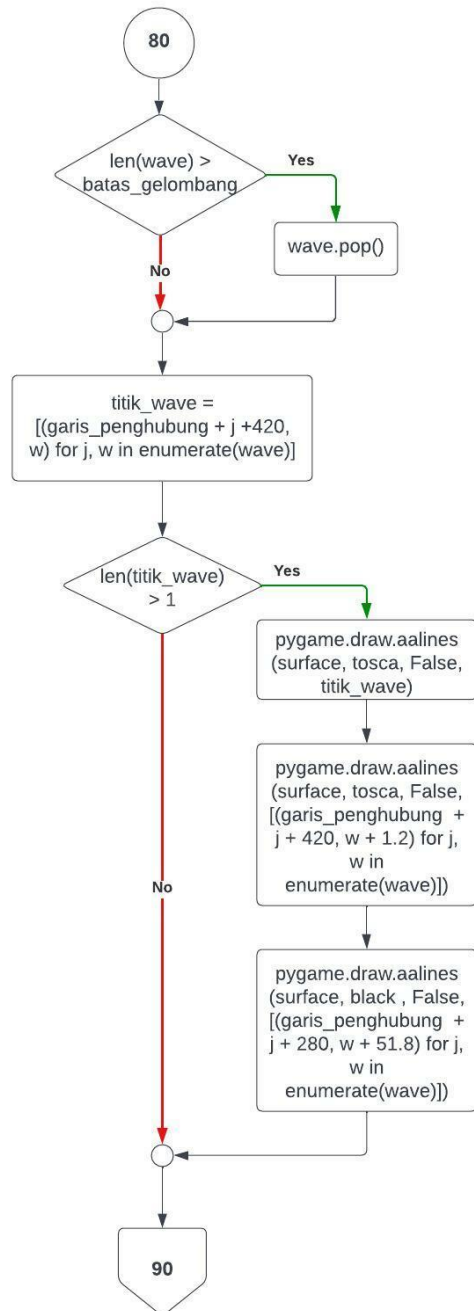
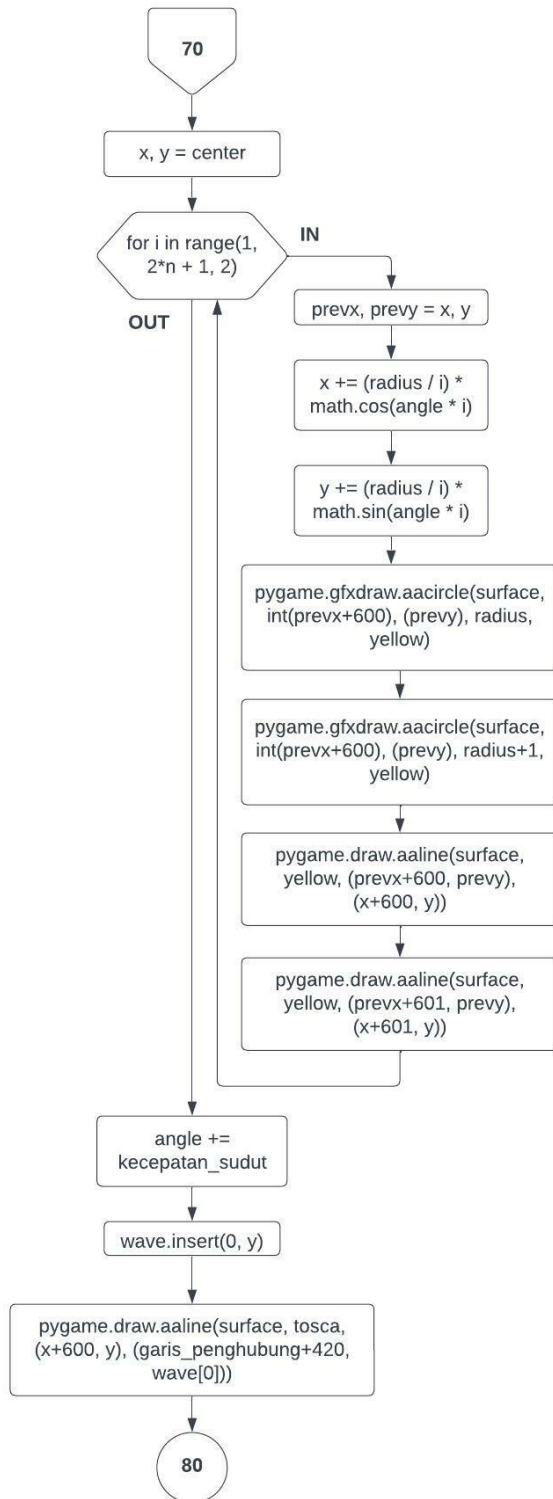
(di halaman selanjutnya)











BAB IV

HASIL DAN PEMBAHASAN

4.1 SOURCE CODE

```
Transverse Wave Using Circle.py - Proyek Kelompok - Visual Studio Code
Transverse Wave Using Circle.py X
Transverse Wave Using Circle.py > ...
1 import pygame, pygame.gfxdraw, pygame.constants, pygame.locals, sys, pygame.locals
2 from pygame import mixer
3 from ctypes import cast, POINTER
4 from comtypes import CLSCTX_ALL
5 from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume
6
7 pygame.init()
8
9 # WARNA
10 yellow = (241,232,94)
11 toska = (89,246,253)
12
13 # LAYAR TAMPLILAN
14 size = 1500, 900
15 surface = pygame.Surface(size)
16 screen = pygame.display.set_mode(size)
17 surface.set_colorkey((0, 0, 0)) # transparent colorkey (warna hitam)
18 pygame.display.set_caption("Transverse Wave Using Circle")
19
20 # KECEPATAN FRAME UNTUK UPDATE
21 FPS = 100
22 clock = pygame.time.Clock()
23
24 # NILAI AWAL
25 pause = False
26 radius = 35
27 kenaikan_radius = 5
28 angle = 0
29 n = 0
30
```

```
Transverse Wave Using Circle.py - Proyek Kelompok - Visual Studio Code
Transverse Wave Using Circle.py X
Transverse Wave Using Circle.py > ...
31 # NILAI AWAL GELOMBANG
32 kecepatan_sudut = 0.005
33 kenaikan_sudut = 0.0049
34 center = 50, 485
35 wave = []
36
37 # PENGATURAN VOLUME - Get default audio device using PyCAW
38 devices = AudioUtilities.GetSpeakers() # menyambungkan perangkat dengan program
39 interface = devices.Activate(IAudioEndpointVolume, iid_, CLSCTX_ALL, None) # interaksi perangkat dengan program
40 volume = cast(interface, POINTER(IAudioEndpointVolume)) # mengatur volume
41 volume.SetMasterVolumeLevel(-65.0, None) # setting suara awal saat program dimulai (-65 sd -3)
42
43 # MUSIK
44 mixer.music.load("sound1.mp3")
45
46 # PANJANG GELOMBANG SAMPAI UJUNG
47 batas_gelombang = 403
48 garis_penghubung = center[0] + 300
49
50 while True:
51     bg = pygame.image.load("bg.jpg").convert()
52     screen.blit(bg, (0, 0))
53     screen.blit(surface, (0, 0))
54     pygame.display.flip()
55     clock.tick(FPS)
56
57     for event in pygame.event.get():
58         if event.type == pygame.QUIT:
59             pygame.quit()
60             volume.SetMasterVolumeLevel(-15.0, None)
```

```
File Edit Selection View Go Run Terminal Help Transverse Wave Using Circle.py - Proyek Kelompok - Visual Studio Code
Transverse Wave Using Circle.py x
Transverse Wave Using Circle.py > ...
61 sys.exit()
62
63 if pygame.key.get_pressed()[pygame.K_p]:
64     mixer.music.play(-1) # looping
65     n = 1
66
67 if pygame.key.get_pressed()[pygame.K_SPACE]:
68     pause = True
69     pygame.mixer.music.pause()
70
71 if pygame.key.get_pressed()[pygame.K_UP]:
72     radius += kenaikan_radius
73     if radius > 70:
74         radius = 70
75
76 if pygame.key.get_pressed()[pygame.K_DOWN]:
77     radius -= kenaikan_radius
78     if radius < 35:
79         radius = 35
80
81 if pygame.key.get_pressed()[pygame.K_RIGHT]:
82     kecepatan_sudut += kenaikan_sudut
83     if kecepatan_sudut > 0.25:
84         kecepatan_sudut = 0.25
85     pygameui.press("volumeup", +1)
86
87 if pygame.key.get_pressed()[pygame.K_LEFT]:
88     kecepatan_sudut -= kenaikan_sudut
89     if kecepatan_sudut < 0.005:
90         kecepatan_sudut = 0.005
```

```
File Edit Selection View Go Run Terminal Help Transverse Wave Using Circle.py - Proyek Kelompok - Visual Studio Code
Transverse Wave Using Circle.py x
Transverse Wave Using Circle.py > ...
91 pygameui.press("volumedown", +1)
92
93 while pause:
94     for event in pygame.event.get():
95         if event.type == pygame.QUIT:
96             pygame.quit()
97             volume.SetMasterVolumeLevel(-15.0, None)
98             sys.exit()
99         if pygame.key.get_pressed()[pygame.K_SPACE]:
100             pause = False
101             pygame.mixer.music.unpause()
102
103 font = pygame.font.Font('Digital-7.ttf', 30)
104 tekskecepatan_sudut = font.render(f"{round(kecepatan_sudut, 10)}", True, toska)
105 amplitudo = font.render(f"{round(radius, 10)}", True, toska)
106
107 surface.fill((0,0,0))
108 surface.blit(tekskecepatan_sudut, (753, 703))
109 surface.blit(amplitudo, (1045, 703))
110
111 x, y = center
112 for i in range(1, 2*n + 1, 2): # (start, stop, step)
113     prevx, prevy = x, y
114     x = x + (radius / i) * math.cos(angle * i)
115     y = y + (radius / i) * math.sin(angle * i)
116
117 # LINGKARAN - aacircle(surface, x, y, r, color)
118 pygame.gfxdraw.aacircle(surface, (prevx+600), (prevy), radius, yellow)
119 pygame.gfxdraw.aacircle(surface, (prevx+600), (prevy), (radius)+1, yellow)
120
```

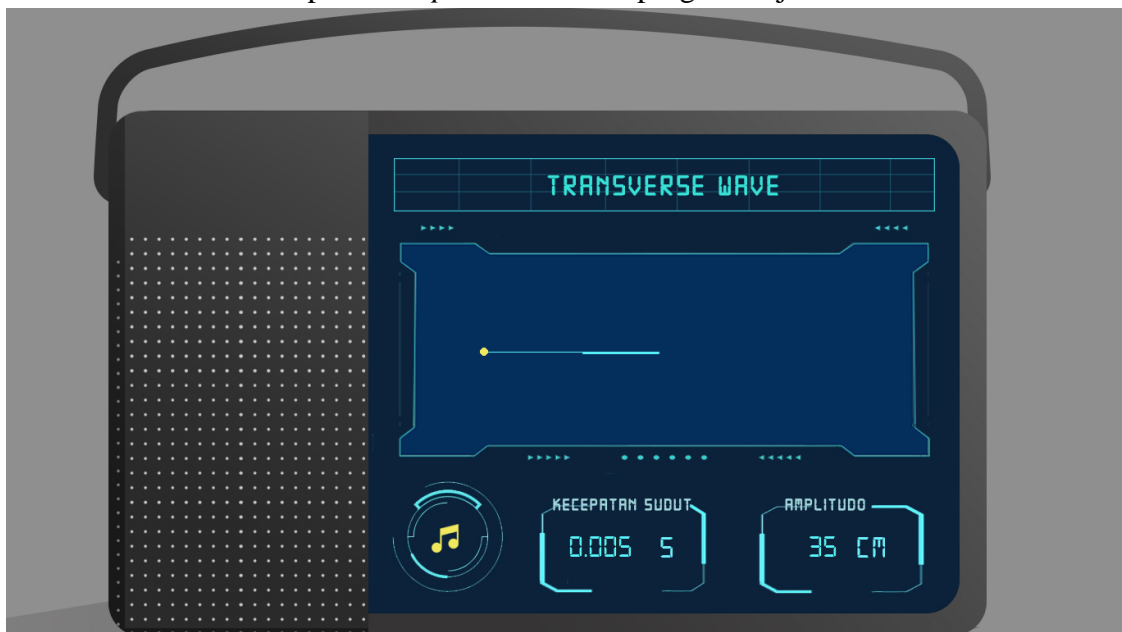
```

File Edit Selection View Go Run Terminal Help Transverse Wave Using Circle.py - Proyek Kelompok - Visual Studio Code
Transverse Wave Using Circle.py
121 # GARIS DALAM LINGKARAN - aaline(surface, color, start_pos, end_pos)
122 pygame.draw.aaline(surface, yellow, (prevx+600, prevy), (x+600, y))
123 pygame.draw.aaline(surface, yellow, (prevx+601, prevy), (x+601, y))
124
125 angle += kecepatan_sudut
126 wave.insert(0, y)
127
128 # GARIS PENUNJUK/PENGHUBUNG - aaline(surface, color, start_pos, end_pos)
129 pygame.draw.aaline(surface, toska, (x+600, y), (garis_penghubung+420, wave[0]))
130
131 # BULATAN PADA GARIS LINGKARAN - circle(surface, color, center, radius)
132 pygame.draw.circle(surface, yellow, [x+600, y], 5)
133
134 if len(wave) > batas_gelombang:
135     wave.pop()
136
137 # POSISI AKHIR
138 titik_wave = [(garis_penghubung + j + 420, w) for j, w in enumerate(wave)] # x, y
139 if len(titik_wave) > 1:
140     pygame.draw.aalines(surface, toska, False, titik_wave)
141     pygame.draw.aalines(surface, toska, False, [(garis_penghubung + j + 420, w + 1.2) for j, w in enumerate(wave)])
142     pygame.draw.aalines(surface, toska, False, [(garis_penghubung + j + 420, w + 1.8) for j, w in enumerate(wave)])

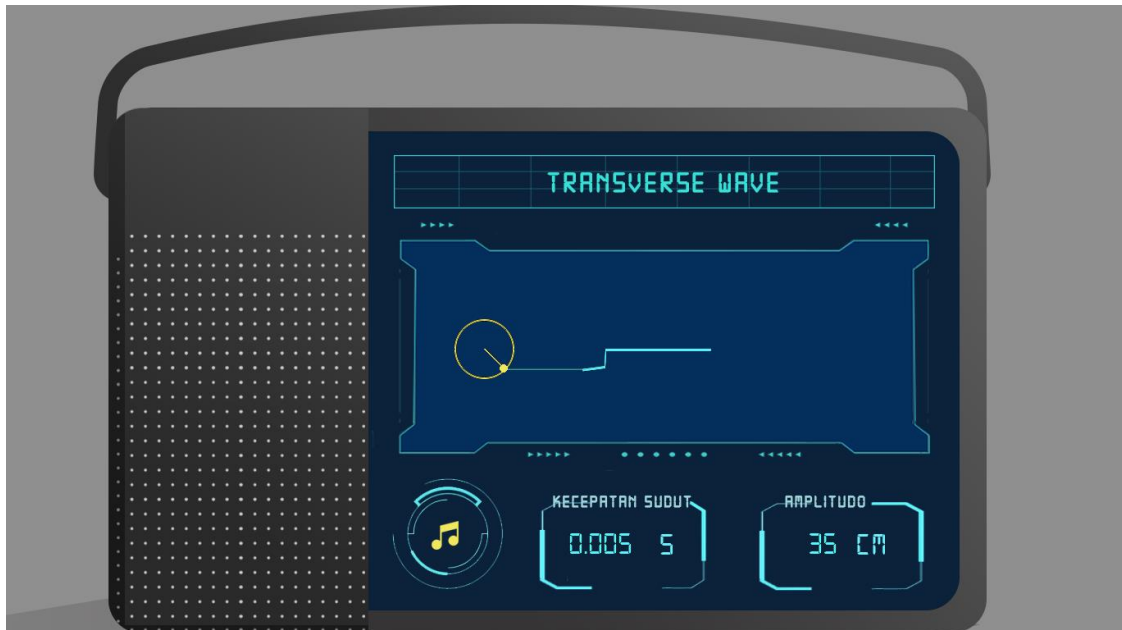
```

4.2 SCREEN CAPTURE HASIL

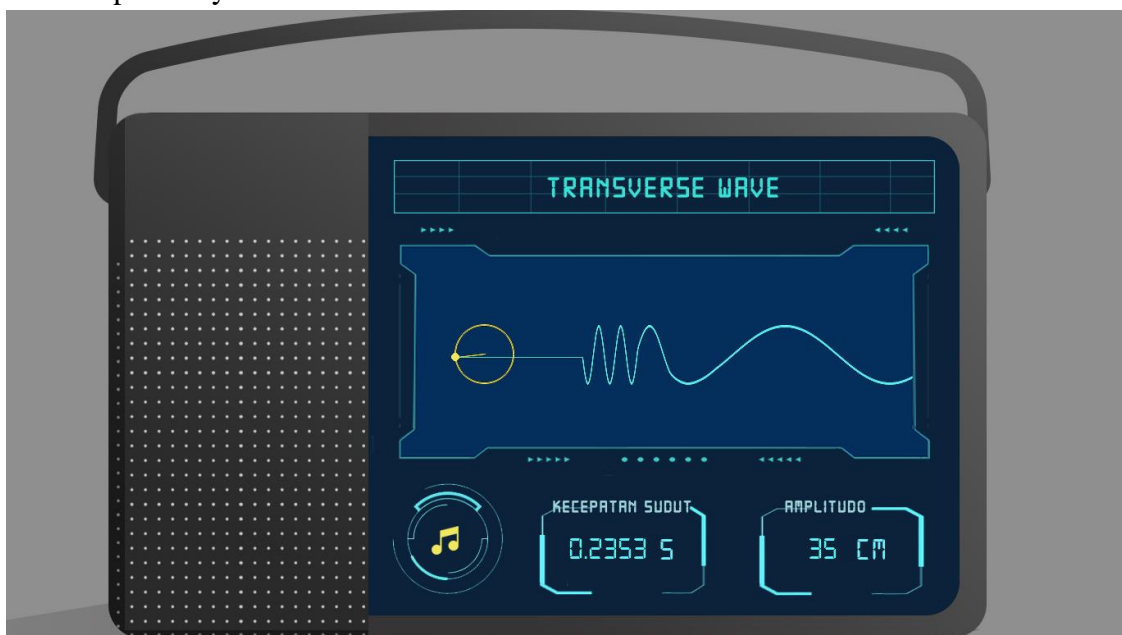
Gambar di bawah merupakan *output* awal ketika program dijalankan.



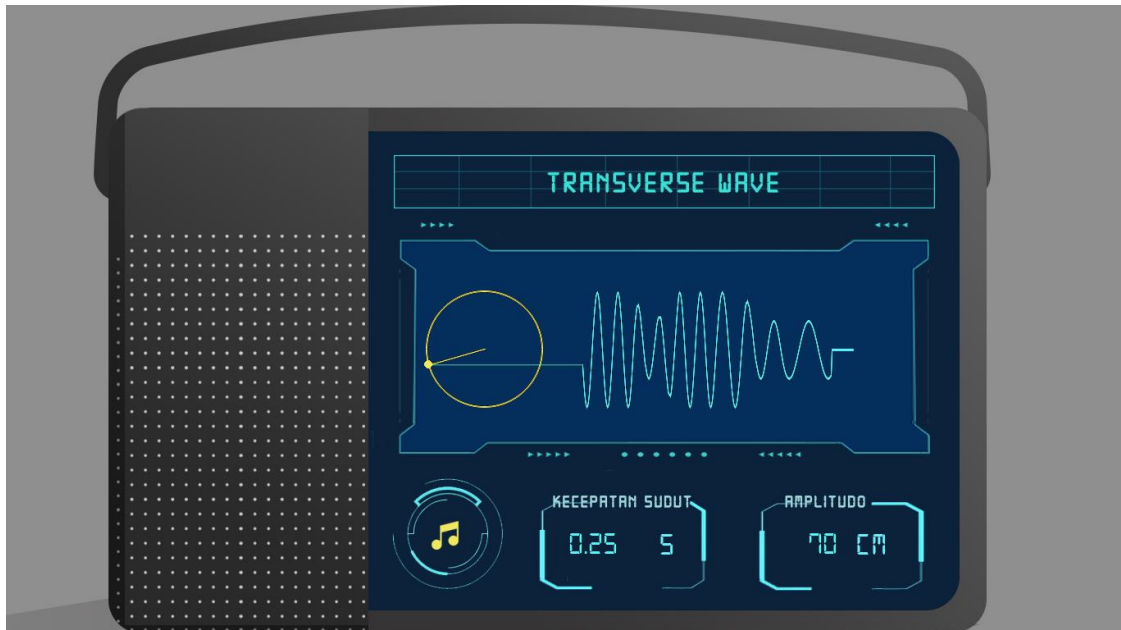
Gambar di bawah merupakan *output* dari lingkaran yang muncul ketika user menekan tombol P pada keyboard.



Gambar di bawah merupakan *output* dari frekuensi gelombang dan kecepatan pergerakan lingkaran bertambah dan berkurang ketika user menekan tombol panah kanan dan kiri pada keyboard.



Gambar di bawah merupakan *output* dari jari - jari dan amplitudo gelombang membesar dan mengecil ketika user menekan tombol panah atas dan bawah pada keyboard.



4.3 PEMBAHASAN

Pada program ini kami mensimulasikan gelombang transversal dengan menggunakan gerak melingkar. Untuk menjalankan program ini dibutuhkan module pygame, sys, pyautogui, math, ctypes, ctypes, dan pycaw. Untuk membuat gelombang tersebut maka dibuatkan lingkaran dengan letak (prevx+600 , prevy) yang dimana merupakan nilai x dan y sebelum diubah yang merupakan letak center. Lingkaran disini memiliki radius sebesar 35 dengan warna yellow. Setelah lingkaran muncul dalam layar maka dibuatlah garis di dalam lingkaran yang menunjukkan gerak melingkar tersebut. Garis ini memiliki letak start post di (prevx+600 , prevy) dan end post di (x+600 , y). Dengan nilai x dan y seperti rumus dibawah ini, dimana nilai x awal bernilai 50 dan nilai y awal bernilai 485. Radius disini bernilai 35, angle bernilai 0 dan i dalam range (1, 2*n + 1, 2).

$$x = x + (\text{radius}/i) * \text{math.cos}(\text{angle}*i)$$

$$y = y + (\text{radius}/i) * \text{math.sin}(\text{angle}*i)$$

Untuk lingkaran dan garis ini kami tebalkan supaya terlihat pada layar dengan menambahkan lingkaran dan garis baru dengan letak yang sama namun menambahkan 1 untuk setiap sumbu x-nya. Setelah lingkaran dan garis muncul pada layar maka dibuatlah garis penghubung antara garis gerak melingkar dengan gelombang. Garis penghubung ini memiliki letak start post pada (x+600, y) dan letak end post pada (garis_penghubung+400

, `wave[0]`). `Garis_penghubung` ini memiliki nilai sebesar `center[0]+300`. Kemudian membuat titik yang menunjukkan gerak melingkar yang terletak pada $(x+600, y)$. Setelah lingkaran, garis gerak melingkar, garis penghubung, dan titik gerak melingkar sudah dapat muncul di layar maka dibuatlah gelombangnya. Untuk membuat gelombang maka kita membuat titik *wave* yang terletak di $(\text{aris_penghubung} + j+420, w)$ dimana *j* dan *w* yang merupakan *enumerate wave*. Apabila `length` dari titik_*wave* tersebut lebih dari 1 maka dibuat garis dengan posisi yang sama dengan titik_*wave* kemudian garis baru dengan posisi $(\text{garis_penghubung} + j+420, w + 1.2)$ dan garis baru lagi dengan letak $(\text{garis_penghubung} + j+420, w + 1.8)$ dimana *j* dan *w* yang merupakan *enumerate wave*.

Tampilan dari gelombang ini kami hubungkan dengan tampilan radio yang menghasilkan musik. Namun tidak terdapat hubungan antara musik atau suara dengan gelombang dalam hal konsep karena suara merupakan gelombang longitudinal sedangkan disini kami membuat simulasi gelombang transversal. Adanya suara disini hanya bertujuan supaya tampilan lebih menarik saja. Supaya interaktif, kami membuat pengguna dapat mengatur kecepatan sudut, amplitudo dan volume. Dimana ketika pengguna menekan ‘p’ pada keyboard maka musik akan mulai dan ketika menekan *directional key* seperti ‘kanan’ maka kecepatan sudut dan volume akan bertambah, selanjutnya apabila menekan ‘kiri’ maka kecepatan sudut dan volume akan berkurang, kemudian apabila menekan ‘atas’ maka amplitudo akan membesar, dan apabila menekan ‘bawah’ maka amplitudo akan mengecil. Ketika pengguna menekan *space* maka simulasi ini akan terhenti sementara dan akan berlanjut lagi ketika *space* ditekan kembali.

Pertambahan kecepatan sudut ini memiliki kenaikan 0.0049 setiap *directional key* ‘kanan’ ditekan dan penurunan 0.0049 setiap *directional key* ‘kiri’ ditekan. Dimana kecepatan sudut awal bernilai 0.005.

$$\text{kecepatan_sudut} = \text{kecepatan_sudut} + \text{kenaikan_sudut}$$

$$\text{kecepatan_sudut} = \text{kecepatan_sudut} - \text{kenaikan_sudut}$$

Pertambahan amplitudo ini memiliki kenaikan 5 setiap *directional key* ‘atas’ ditekan dan penurunan 5 setiap *directional key* ‘bawah’ ditekan. Amplitudo disini sama dengan radius yang memiliki nilai awal 35. Dan apabila radius sudah mencapai 70 maka sistem akan memberhentikan penambahan radius.

$$\text{radius} = \text{radius} + \text{kenaikan_radius}$$

$$\text{radius} = \text{radius} - \text{kenaikan_radius}$$

BAB V

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Simulasi kali ini membentuk gelombang transversal yang merupakan gabungan dari gerak melingkar yang terbentuk dari amplitudo (A) dan kecepatan sudut (ω) dengan konsep gerak harmonis dan teori gelombang. Gelombang terbentuk dari pergerakan lingkaran yang equilibrium dan amplitudo gelombang dipengaruhi oleh besar kecilnya jari - jari lingkaran. Semakin besar jari-jari pada lingkaran, maka amplitudo gelombang yang terbentuk akan ikut membesar yang artinya gelombang menjadi semakin tinggi atau lebar. Selain itu, semakin diperbesar kecepatan sudutnya maka pembentukan dari gelombang akan semakin cepat. Apabila diperkecil kecepatan sudutnya, maka pembentukan gelombang akan melambat. Gelombang pada simulasi ini dapat terbentuk melalui plot dari titik-titik x dan y pada list titik_wave. Musik pada simulasi gelombang transversal yang terbentuk dari pergerakan gerak melingkar hanya untuk mempercantik simulasi, agar simulasi tidak membosankan, tidak ada hubungannya dengan simulasi karena bunyi merupakan gelombang longitudinal bukan gelombang transversal.

5.2 SARAN

5.21 Saran dari kelompok 8

Simulasi dapat dibuat dari konsep yang sederhana agar dapat dimengerti dengan mudah oleh mahasiswa. Selain itu, penggabungan konsep juga bisa menjadi pertimbangan untuk membuat simulasi agar lebih menarik.

5.22 Saran dari Penguji

Sebaiknya volume musik tidak usah dibuat membesar dan mengecil karena dikhawatirkan akan terjadi miskonsepsi.

DAFTAR PUSTAKA

French, A. P. (2017). *Vibrations and waves*, Florida: CRC press.

Giancoli, D, C (2004). *Physics Principles with Application Sixth Edition*, New Jersey: Pearson Education.

Handika, F. & Fatahillah, D. A. (2019). Bentuk polinom gelombang transversal dengan pembuktian deret taylor dengan sisa. *Prosiding Seminar Pendidikan Matematika dan Matematika*, 1, 23-28.

King, G. C. (2009). *Vibrations and Waves 1st edition*. Chichester, U.K: Wiley.

Li, J., Liu, X., Chen, Z., & Wan, P. (2020). Design of a New Audio Signal Generation Circuit. *2020 IEEE 14th International Conference on Anti-Counterfeiting, Security, and Identification (ASID)*, 154–158.

LAMPIRAN

Sumber coding sebagai referensi pada pemrograman kami berasal dari https://github.com/shelldoom/fourier_series. Dalam sumber ini dibahas fourier series dimana lingkaran dapat membentuk gelombang sin dan gelombang square atau kotak seperti pada gambar berikut

