

LAPORAN PROJEK II

“ Sistem Kontrol *Air Conditioner* (AC) Menggunakan Pendekatan Berpikir Komputasional“

MATA KULIAH COMPUTATIONAL THINKING (WI1102)

TAHUN 2024



Disusun oleh:

Rhenaldy Cahyadi Putra	19624019
Suryani Mulia Utami	19624020
Muhammad Haris Putra S.	19624032
Miguel Rangga Deardo Sinaga	19624053
Muchammad Rafif Azis Syahlevi	19624061

Kelompok 7

Kelas 28

**TUGAS BESAR MATA KULIAH WAJIB KURIKULUM
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA - KOMPUTASI
INSTITUT TEKNOLOGI BANDUNG
2024**

SURAT PERNYATAAN PENGGUNAAN KECERDASAN BUATAN/AI

Dengan ini, **saya/kami***:

Kelompok/Kelas	:	7/K-28
Fakultas/Sekolah	:	STEI
Kode Mata Kuliah	:	WI1102
Nama Mata Kuliah	:	Computational Thinking (CT)
Judul Tugas/Proposal	:	Sistem Kontrol Air Conditioner (AC) Menggunakan Pendekatan Berpikir Komputasional

Menyatakan bahwa **saya/kami*** menggunakan/tidak menggunakan* AI/_kecerdasan buatan dalam pengerjaan maupun penyusunan luaran tugas pada mata kuliah yang tertulis di atas.

Jika Ya, maka alat AI/kecerdasan buatan yang digunakan adalah:

Lingkup Pekerjaan	Digunakan?	Tingkat Penggunaan (level 1-5)
Pemeriksaan Ejaan: menggunakan tools seperti Grammarly, DeepL, Quillbot, Grammarbot, LanguageTool, ProWritingAid, ChatGPT, Google Gemini, atau sejenisnya. Sebutkan tool yang digunakan:	Ya/Tidak*	0
Pembuatan Teks: menggunakan tools seperti ChatGPT, Gemini, Copilot, GrammarlyGO, WordAI, WriteSonic, Jasper, Jenni AI, atau sejenisnya. Sebutkan tool yang digunakan:	Ya/Tidak*	0
Bantuan Diskusi dalam Penyusunan Konten: menggunakan tools seperti ChatGPT, Gemini, Copilot, Perplexity, Jenni AI, Zoom Companion, atau sejenisnya. Sebutkan tool yang digunakan:	Ya/Tidak*	2
Pembuatan Gambar, Video, dan/atau Grafik: menggunakan tools seperti Craiyon, DALL-E, Midjourney, Stable Diffusion, Microsoft Designer, Gemini, Canva AI, atau sejenisnya. Sebutkan tool yang digunakan:	Ya/Tidak*	0

Saya/Kami* juga menyatakan bahwa setiap penggunaan AI/kecerdasan buatan yang dilakukan pada mata kuliah tersebut telah dinyatakan di dalam surat ini.

DAFTAR ISI

BAB 1	2
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	1
1.3. Tujuan Penelitian	1
1.4. Manfaat Penelitian	2
BAB 2	2
KAJIAN TEORI	2
2.1. Computational Thinking.....	2
2.2. Pemrograman	4
BAB 3	4
ISI	4
3.1. Cara Kerja Singkat AC.....	4
3.2. Dekomposisi Sistem AC	5
3.3. Abstraksi Sistem AC.....	7
3.4. Pengenalan Pola Sistem AC	7
3.5. Algoritma Sistem AC	9
3.6. Implementasi Program	15
BAB 4	23
KESIMPULAN.....	23
DAFTAR PUSTAKA	24
PEMBAGIAN KERJA	24

BAB 1

BAB 1

PENDAHULUAN

1.1.Latar Belakang

Sistem kerja AC yang kompleks memerlukan pendekatan dari sudut pandang lain agar dapat dipahami dengan lebih mudah. Pendekatan tersebut harus mampu membantu menyederhanakan, memahami, dan menyelesaikan permasalahan sistem AC secara sistematis. Salah satu pendekatan yang memungkinkan adalah dengan *Computational Thinking* (CT). CT memungkinkan kita untuk dapat memecah masalah besar menjadi bagian-bagian kecil yang lebih mudah dipahami dan dikelola. Pendekatan ini sangat penting ketika menghadapi sistem benda atau mesin, seperti AC, yang memiliki banyak komponen dan proses. Dengan memecahnya menjadi bagian-bagian kecil dan komponen-komponen sederhana, kita dapat mengenali pola sistem, serta berfokus pada komponen utama yang merepresentasikan sistem AC. Selain itu, CT juga mendukung perancangan logika atau algoritma yang dapat diimplementasikan pada sistem benda atau mesin, seperti pada pengaturan suhu dan kelembapan dalam sistem AC.

Teknologi AC terus mengalami inovasi, mulai dari sistem pendingin yang hemat energi hingga pengaturan suhu yang cerdas dan ramah lingkungan. Di balik layar, terdapat berbagai teknologi canggih yang mengatur setiap aspek dari cara kerja AC, seperti kompresor *inverter*, refrigeran ramah lingkungan, hingga algoritma otomatis yang mampu membaca dan menyesuaikan kondisi ruangan. Perangkat ini tidak hanya membantu mengontrol suhu, tetapi juga memengaruhi kualitas udara, kelembapan, hingga kesehatan lingkungan tempat kita tinggal.

1.2.Perumusan Masalah

- Bagaimana penerapan *Computational Thinking* untuk menyederhanakan proses kerja sistem pendingin udara?
- Apa saja komponen utama dan pola kerja sistem pendingin udara yang dapat diidentifikasi dan direpresentasikan secara sederhana?
- Bagaimana merancang algoritma yang mampu merepresentasikan fungsi dasar sistem pendingin udara?

- Bagaimana membuat program fungsi dasar sistem pendingin udara berdasarkan rancangan algoritma yang telah dibuat?

1.3.Tujuan Penelitian

- Mengimplementasikan konsep *Computational Thinking* pada kasus dunia nyata.
- Menyederhanakan proses kerja sistem pendingin udara yang kompleks agar lebih mudah dipahami dengan menggunakan pendekatan *Computational Thinking*.
- Mengidentifikasi komponen utama dan pola kerja sistem pendingin udara yang dapat direpresentasikan secara sederhana dalam sebuah model.
- Merancang algoritma sederhana yang dapat merepresentasikan fungsi dasar sistem pendingin udara, seperti pengaturan suhu dan kelembapan.
- Membuat program fungsi dasar dari sistem pendingin udara berdasarkan rancangan algoritma yang telah dibuat.

Pembaruan :

- Mengidentifikasi komponen utama dan pola kerja sistem pendingin udara yang dapat direpresentasikan secara sederhana dalam sebuah model UI remote AC.
- Merancang algoritma sederhana yang dapat merepresentasikan fungsi dasar sistem pada remote AC pendingin udara, seperti Tombol On/Off, Mode, Timer, Speed, dan lainnya.

1.4.Manfaat Penelitian

- Memberikan pemahaman yang lebih dalam mengenai konsep *Computational Thinking* dan penerapannya dalam menyederhanakan sistem yang kompleks dan nyata.
- Mampu mengimplementasikan konsep *Computational Thinking* pada kasus dunia nyata.
- Membiasakan diri setiap anggota kelompok untuk menerapkan konsep *Computational Thinking* dalam konteks yang relevan pada kehidupan sehari-hari.
- Mengembangkan keterampilan dalam merancang algoritma sederhana yang dapat diterapkan pada berbagai sistem teknologi, khususnya sistem pendingin udara (AC).

- Mengembangkan keterampilan dalam membuat program fungsi dasar suatu sistem, khususnya sistem pendingin udara (AC) berdasarkan rancangan algoritma yang telah dibuat.

BAB 2

KAJIAN TEORI

2.1. Computational Thinking

Kemampuan berpikir kreatif, kritis, komunikasi, dan kolaborasi merupakan keterampilan utama yang sangat penting dalam pembelajaran abad ke-21, di samping kemampuan dasar seperti literasi membaca, matematika, dan sains. Salah satu kemampuan berpikir yang semakin relevan untuk dikuasai adalah *Computational Thinking* (CT). CT didefinisikan sebagai proses berpikir untuk merumuskan masalah dan solusinya dengan cara yang memungkinkan solusi tersebut diimplementasikan secara efektif oleh agen pemroses informasi, seperti komputer, robot, atau bahkan manusia.

Computational Thinking (CT) adalah pendekatan berpikir yang digunakan untuk memecahkan masalah, merancang sistem, dan memahami perilaku manusia. CT menjadi landasan dari konsep informatika. Di era modern, di mana komputer telah menjadi bagian integral dalam berbagai aspek kehidupan, CT perlu dijadikan dasar dalam cara seseorang berpikir dan memahami dunia yang penuh dengan tantangan kompleks. CT melibatkan proses berpikir untuk menciptakan dan menerapkan berbagai tingkat abstraksi, mulai dari memahami masalah hingga menghasilkan solusi yang efektif, efisien, adil, dan aman. Selain itu, CT juga mencakup pemahaman terhadap dampak dari skala masalah dan tingkat kompleksitasnya, tidak hanya untuk alasan efisiensi tetapi juga untuk pertimbangan ekonomi dan sosial.

Computational Thinking (CT) melibatkan metode dan proses berpikir untuk menyelesaikan masalah dengan menerapkan langkah-langkah berikut:

- Dekomposisi persoalan: Menyelesaikan sebuah masalah dengan cara membagi masalah menjadi sub-masalah yang lebih kecil, kemudian merumuskan masalah-masalah yang lebih kecil tersebut secara terpisah.
- Abstraksi: Mengidentifikasi elemen-elemen penting dari suatu masalah dengan mengabaikan hal-hal yang kurang relevan, sehingga memudahkan fokus pada solusi.

- Pengenalan pola: Mengidentifikasi kesamaan/perbedaan karakteristik dari berbagai permasalahan atau fenomena yang berbeda tetapi memiliki keterkaitan.
- Algoritma: Merancang solusi secara otomatis melalui pemikiran algoritmik, yaitu menyusun langkah-langkah yang terstruktur dan sistematis. Dalam algoritma, terdapat konsep-konsep dasar yang membantu membangun solusi pemrograman:
 - Sekuens
Sekuens adalah urutan langkah-langkah yang dijalankan satu per satu dari awal hingga akhir. Setiap langkah dilakukan secara berurutan tanpa ada pengulangan atau percabangan.
 - Selection
Selection digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Tergantung pada hasil dari kondisi tersebut, program akan menjalankan satu dari beberapa jalur yang ada.
 - Repetisi
Repetisi adalah proses menjalankan satu atau beberapa langkah secara berulang hingga kondisi tertentu terpenuhi.
 - Array
Array adalah struktur data yang digunakan untuk menyimpan sejumlah elemen dengan tipe data yang sama, dalam satu variabel. Elemen-elemen array diakses menggunakan indeks.

Konsep-konsep ini adalah dasar dalam menyusun algoritma yang efisien untuk menyelesaikan masalah dengan bantuan komputer.

2.2.Pemrograman

Pemrograman adalah proses merancang dan menulis serangkaian instruksi (kode) yang dijalankan oleh komputer untuk melakukan tugas tertentu. Kegiatan ini melibatkan penerjemahan solusi masalah ke dalam bentuk algoritma dan kode program menggunakan bahasa pemrograman. *Programmer* mengubah algoritma menjadi kode sumber (*source code*) yang kemudian diolah oleh *compiler* atau *interpreter* agar dapat dipahami oleh komputer. Pemrograman memungkinkan komputer menyelesaikan berbagai masalah, mulai dari yang sederhana hingga kompleks, dengan efisien dan terstruktur.

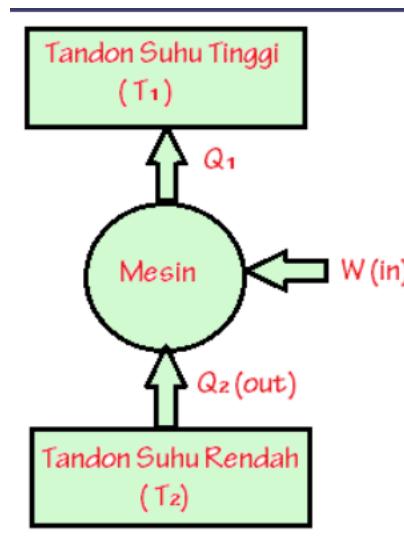
Komputer sendiri memproses instruksi dalam bahasa mesin (*machine language*), yang berupa kode numerik untuk menjalankan operasi dasar seperti penjumlahan, pengurangan, perbandingan, pemindahan data, atau pengulangan instruksi. Namun, *programmer* lebih sering menggunakan bahasa pemrograman tingkat tinggi (*high-level languages*) untuk menulis kode program, seperti C/C++, Python, Matlab, Scratch, dan lain-lain.

BAB 3

ISI

3.1.Cara Kerja Singkat AC

AC bekerja sebagai mesin pendingin Carnot. Carnot adalah suatu mesin kalor yang bekerja dalam suatu siklus yang dinamakan siklus Carnot. Gaya eksternal yang diberlakukan pada mesin dapat menyebabkan pemindahan kalor dari tempat yang dingin ke tempat yang lebih panas.



Mengambil panas dari udara dalam ruangan

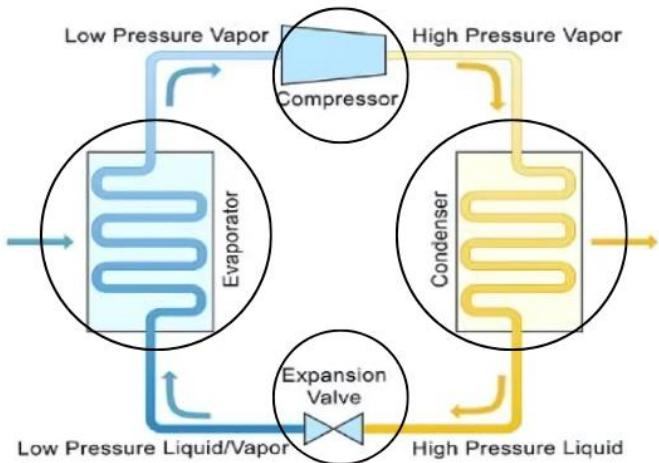
|
v
Membuat udara menjadi dingin

|
v
Menyirkulasikannya kembali ke dalam ruangan

3.2. Dekomposisi Sistem AC

Sebagai mesin pendingin Carnot atau mesin refrigerasi yang mengadopsi kebalikan dari siklus Carnot, pada AC perlu diberikan kerja atau gaya eksternal untuk ‘memaksa’ kalor mengalir dari benda dingin ke benda panas.

Berdasarkan perannya dalam melakukan siklus kerja AC, komponen AC dapat dibagi menjadi 4 komponen utama dan 1 komponen pendukung, sebagai berikut:



A. Evaporator

Merupakan tempat terjadinya penyerapan panas dari udara sekitar oleh refrigeran sehingga udara sekitar menjadi dingin. Dalam hal ini fungsi evaporator adalah menyediakan permukaan yang memungkinkan pertukaran panas antara udara dan refrigeran.

B. Kompresor

Sesuai namanya, fungsinya adalah meningkatkan tekanan gas refrigeran yang telah menyerap panas di evaporator. Dengan memampatkannya, gas refrigeran kini memiliki tekanan dan suhu yang sangat tinggi. Tujuannya adalah untuk meningkatkan energi dalam gas refrigeran agar dapat melepaskan panas dengan lebih efektif di tahap selanjutnya.

C. Kondensor

Merupakan tempat di mana terjadi pelepasan kalor gas refrigeran dari kompresor yang bersuhu tinggi ke udara luar. Dengan begitu, refrigeran tidak bersuhu tinggi lagi.

D. Katup Ekspansi

Berfungsi untuk menurunkan tekanan gas refrigeran yang sebelumnya tinggi karena dikompresi, dan saat yang bersamaan, sesuai dengan hukum termodinamika, suhu gas refrigeran juga akan ikut menurun. Dengan begitu, gas refrigeran bersuhu rendah kembali dan siap untuk menyerap panas di evaporator.

E. Fan

Fan merupakan komponen penting pendukung kerja sistem AC. Di evaporator, *evaporator fan* membantu untuk mendistribusikan udara dingin ke seluruh ruangan. Sedangkan di kondensor, *condenser fan* membantu melepaskan kalor refrigeran ke udara luar.

3.3. Abstraksi Sistem AC

Sistem pendingin ruangan (AC) adalah perangkat yang dirancang untuk menjaga suhu ruangan tetap nyaman dengan memanfaatkan berbagai komponen yang saling berinteraksi, seperti sensor suhu, kompresor, dan kipas. Dalam memodelkan permasalahan ini, abstraksi dilakukan dengan memfokuskan pada fungsi utama sistem, yaitu menjaga suhu ruangan sesuai dengan nilai yang diinginkan, sambil mempertimbangkan efisiensi energi dan responsivitas sistem terhadap perubahan kondisi lingkungan.

Proses *Computational Thinking* dalam abstraksi ini melibatkan penyederhanaan aspek-aspek teknis dengan cara mengelompokkan fitur-fitur inti dan menghilangkan detail yang tidak relevan, seperti sifat fisik material tertentu, variasi komponen minor, dan fitur seperti *heater* karena kurang relevan di Indonesia karena alasan iklim tropis. Fokus utama kami adalah fitur *temperature*, *air-swing*, dan *fan-speed* yang merupakan inti utama pendingin ruangan (AC). Dengan demikian, sistem dapat direpresentasikan sebagai hubungan antara *input* (suhu aktual dan suhu yang diinginkan), mekanisme proses, dan *output* (kinerja sistem dalam mempertahankan suhu).

Lalu pada proses penyusunan algoritma, kami akan mengabaikan komponen - komponen pendingin ruangan (AC) yang statis atau tidak bisa dapat diatur oleh pengguna.

Contoh komponen statis (tidak perlu algoritma pemrograman) adalah Evaporator, Kondensor, dan Katup Ekspansi. Sedangkan komponen dinamis (perlu diatur pengguna dan perlu algoritma) adalah Sensor Suhu, *Fan Indoor*, *Fan Outdoor*, dan Kompresor.

3.4. Pengenalan Pola Sistem AC

A. Tombol Power

- Fungsi Utama:

Tombol *power* adalah tombol yang paling sering digunakan pada *remote AC*, yang berfungsi untuk menyalakan atau mematikan unit AC. Penggunaannya sangat mendasar dan esensial karena tanpa tombol ini, semua pengaturan lainnya pada *remote* menjadi tidak berguna.

- Pola Penggunaan:

Kebiasaan pengguna menunjukkan bahwa tombol *power* digunakan setiap kali seseorang memasuki ruangan atau meninggalkan ruangan, sebagai langkah pertama dan terakhir dalam penggunaan AC. Sebagian besar orang akan menyalakan AC saat memasuki ruang untuk mendapatkan kenyamanan suhu, dan mematikannya ketika keluar atau saat suhu ruangan telah cukup nyaman. Frekuensi penggunaan tombol ini sangat tinggi, dengan banyak pengguna menyalakan atau mematikan AC lebih dari sekali dalam sehari.

B. Tombol Temperatur (Suhu)

- Fungsi Utama:

Tombol temperatur berfungsi untuk mengatur suhu ruangan dengan menaikkan atau menurunkan suhu sesuai dengan keinginan pengguna. Pengaturan suhu menjadi sangat penting dalam menjaga kenyamanan individu, baik untuk tidur, bekerja, atau aktivitas lainnya di dalam ruangan.

- Pola Penggunaan:

Penggunaan tombol ini umumnya lebih sering terjadi saat cuaca di luar ruangan berubah drastis atau saat pengguna ingin menyesuaikan kenyamanan pribadi. Pengguna akan menurunkan suhu ketika cuaca panas atau pada siang hari yang terik, dan menaikkan suhu saat cuaca dingin atau pada malam hari. Oleh karena itu, tombol temperatur memiliki penggunaan yang lebih fleksibel berdasarkan kondisi eksternal dan preferensi pribadi.

C. Tombol Mode

- Fungsi Utama:

Tombol mode memberikan opsi bagi pengguna untuk memilih antara beberapa mode operasional AC, seperti *Cool*, *Fan*, dan *Dry*. Masing-masing mode ini memiliki fungsi yang berbeda untuk menyesuaikan kebutuhan pengguna:

- Mode *Cool*: Menggunakan sistem pendinginan untuk menurunkan suhu ruangan.
 - Mode *Fan*: Menggunakan kipas untuk sirkulasi udara tanpa pendinginan.
 - Mode *Dry*: Mengurangi kelembapan di ruangan tanpa perubahan suhu yang signifikan.
 - Mode *Turbo* : AC langsung memaksimalkan kecepatan kipas dan pendinginan atau pemanasan secara instan untuk mencapai suhu yang diinginkan lebih cepat.
 - Mode *Sleep* : AC menaikkan suhu target secara berkala dan dapat mempertahankan suhu.
 - Mode *Auto* : AC secara otomatis menyesuaikan mode, suhu, dan kecepatan kipas berdasarkan suhu ruangan untuk mencapai kenyamanan optimal.
 - Mode *Eco* : AC memprioritaskan efisiensi energi daripada pendinginan atau pemanasan cepat. Konsumsi daya jauh lebih rendah.
-
- Pola Penggunaan:

Pengguna umumnya memilih mode *Cool* pada siang hari yang panas untuk pendinginan cepat, sementara mode *Fan* lebih sering dipilih pada malam hari atau ketika suhu ruangan sudah cukup nyaman dan pengguna hanya membutuhkan sirkulasi udara. Mode *Dry* lebih sering dipilih pada musim hujan atau daerah dengan kelembapan tinggi, di mana pengguna lebih membutuhkan dehumidifikasi udara. Automasi Mode: Implementasi mode otomatis yang dapat menyesuaikan pengaturan mode berdasarkan suhu dan kelembapan ruangan bisa sangat berguna. Misalnya, jika kelembapan udara terdeteksi tinggi, AC secara otomatis akan beralih ke mode *Dry*.

D. Tombol Fan Speed

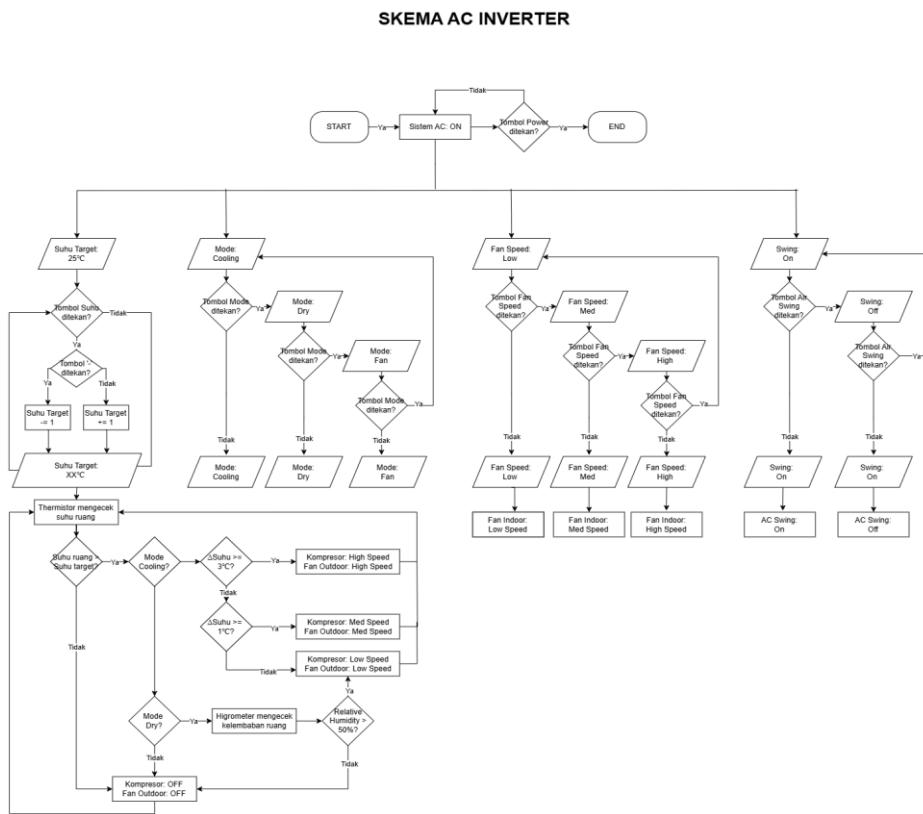
- Fungsi Utama:

Tombol *fan speed* digunakan untuk mengatur kecepatan kipas dalam unit AC. Pengguna dapat memilih antara *Low*, *Medium*, atau *High* untuk menyesuaikan tingkat sirkulasi udara di ruangan.

- Pola Penggunaan:

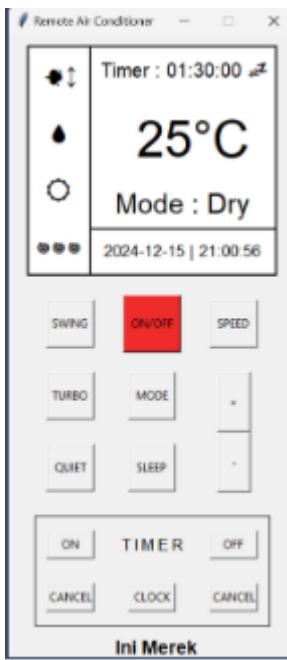
Bergantung pada waktu penggunaan, tombol ini lebih sering digunakan saat pengguna membutuhkan tingkat kebisingan yang lebih rendah (misalnya, saat tidur atau bekerja). Pada malam hari, banyak pengguna memilih *fan speed* rendah untuk mengurangi kebisingan dari kipas. Sebaliknya, pada siang hari yang terik, pengguna cenderung memilih *fan speed* tinggi untuk memastikan pendinginan yang lebih cepat.

3.5.Flowchart Sistem AC



3.6.Algoritma Sistem AC

- Graphic User Interface



Alasan penerapan GUI pada remote AC: Lebih mudah memvisualisasikan remote AC seperti di dunia nyata, lebih mudah dipahami. Efisiensi dalam pembuatan program. Meningkatkan daya tarik program. Sesuai dengan standar perangkat keras saat ini. Karena hampir semua perangkat elektronik menggunakan antarmuka berbasis GUI termasuk remote AC.

- Library Tkinter

```

1 import tkinter as tk
2 from tkinter import *
3
4 import datetime as dt
5 from datetime import datetime
6
7 # Jendela Utama
8 lebar=100
9 tinggi=650
10
11 window=tk.Tk()
12 lebarlayar=window.winfo_screenwidth()
13 tinggilayar=window.winfo_screenheight()
14
15 xint((lebarJayar/2)-(lebar/2))
16 y-int((tinggilayar/2)-(tinggi/2))-30
17
18 window.geometry(f'{lebar}x{tinggi}+{xint}+{y}')
19 window.resizable(0,0)
20 window.title("Remote Air Conditioner")
21
22 # ----- LAYAR REMOTE AC -----
23
24 # Membuat layar dengan garis tepi
25 layer = tk.Frame(window, bg="white", width=100, height=650)
26 layer.place(x=20,y=20)
27
28 # Membuat layer 1 pada layer
29 layer1 = tk.Frame(layer, bg="white", width=100, height=650)
30 layer1.place(x=35,y=10)
31
32 # Membuat layer 2 pada layer
33 layer2 = tk.Frame(layer, bg="white", width=100, height=650)
34 layer2.place(x=35,y=10)

```

Library Tkinter adalah pilihan ideal untuk membuat aplikasi GUI seperti remote AC karena: Mudah digunakan. Tidak memerlukan instalasi tambahan (library bawaan python) Ringan dan fleksibel. Cocok untuk edukasi dan prototyping program. Mendukung integrasi dengan library lain untuk fungsi canggih seperti IoT.

- Library Pendukung

```
# Menampilkan waktu real-time
current_time = datetime.now()
# Format waktu agar lebih rapi
formatted_time = current_time.strftime("%Y-%m-%d | %H:%M:%S")

import datetime as dt
from datetime import datetime
```



Library datetime untuk menampilkan waktu secara real-time sekaligus untuk konfigurasi mode sleep

Pada program remote AC ini kami juga berencana untuk menambah memori jangka pendek pada AC, mode timer, merubah font menjadi seven-bar digit yang dimana kemungkinan besar akan menambah library lain pada source code kami.

- Fitur Memory

Fitur Memory adalah fitur yang menyimpan suhu terakhir sebelum remote ac dimatikan Hal ini dapat dilakukan dengan mengimport library OS. Library ini berfungsi untuk berinteraksi dengan OS atau operating system. Dengan menggunakan library ini, suhu terakhir dari AC dapat disimpan melalui file txt yang nantinya akan disimpan dan dapat dibuka kembali saat menyalakan remote AC.

- Fitur Timer

```
# Fungsi untuk memperbarui label timer
def update_timer_label():
    minutes, seconds = divmod(remaining_time, 60)
    timer_display = f"{minutes:02d}:{seconds:02d}"
    label_timer.config(text=timer_display)

# Mengambil waktu awal
def start_time():
    global remaining_time
    try:
        # Meminta pengguna masukkan waktu dalam menit
        time_minutes = simpledialog.askstring("Set Timer", "Masukkan waktu (dalam menit):")
        if time_minutes is None or time_minutes <= 0:
            return
        remaining_time = int(time_minutes) * 60
        update_timer_label()
        count_time()
    except ValueError:
        print("Input tidak valid. Masukkan angka saja.")

# Mengontrol timer
def count_time():
    global remaining_time
    if remaining_time > 0:
        update_timer_label()
        remaining_time -= 1
        window.after(1000, count_time)
    else:
        label_timer.config(text="Waktu habis!")
        stop_time()
```

Fitur Timer pada kontrol AC seperti yang kita ketahui berguna untuk mematikan AC sesuai dengan rentang waktu yang diinginkan user.

Note : Perlu menentukan variable waktu awal. Pada code ini variable tersebut adalah remaining_time Perlu mengimport module simpledialog dari library tkinter untuk menjalankan dialog yang berfungsi sebagai input timer

A. Urutan (*Sequence*)

Sekuens adalah bagian dari kode yang dieksekusi secara berurutan tanpa adanya percabangan atau pengulangan. Ini berarti instruksi dalam bagian sekuens dieksekusi satu per satu dalam urutan yang sudah ditentukan.

Contoh dalam code:

- Program 1: Remote Air Conditioner Universal

```
ac_target_code = "006"
synced_code = auto_sync(ac_target_code)

if synced_code:
    print(f"Sinkronisasi selesai. Remote siap digunakan dengan kode: {synced_code}")
```

- Program 2: Remote Air Conditioner GUI

```
lebar = 300
tinggi = 650

window = tk.Tk()
lebarlayar = window.winfo_screenwidth()
tinggilayar = window.winfo_screenheight()
```

Ini adalah bagian sekuens karena variabel-variabel tersebut diinisialisasi secara langsung tanpa kondisi atau *loop*.

- **Fungsi simpan suhu:** Fungsi ini digunakan untuk menyimpan suhu ke file

```
# Fungsi untuk menyimpan suhu ke file
def simpan_suhu(suhu):
    with open("suhu_terakhir.txt", "w") as file:
        file.write(str(suhu))
```

Fungsi ini berjalan tanpa kondisi pengulangan, dan hanya dieksekusi berdasarkan status saat ini. Ini adalah contoh sekuens.

- **Fungsi baca suhu:** Fungsi ini digunakan untuk membaca suhu dari file

```
# Fungsi untuk membaca suhu dari file
def baca_suhu():
    try:
        with open("suhu_terakhir.txt", "r") as file:
            return int(file.read())
    except FileNotFoundError:
        return 24 # Suhu default jika file tidak ditemukan
```

Ini juga termasuk dalam sekuens, karena instruksi yang ada dijalankan secara berurutan.

B. Percabangan (*Selection*)

Selection adalah bagian dari kode yang membuat program membuat keputusan berdasarkan kondisi tertentu. Kondisi ini akan mengarahkan alur program ke bagian tertentu berdasarkan nilai yang diberikan.

Contoh dalam *code*:

- **Pengecekan suhu:** Bagian ini mengecek suhu untuk mode heat dan cold

```
if suhu > 30:  
    suhu = 16  
    heat = True  
elif suhu < 10:  
    suhu = 18  
    cold = True
```

Bagian ini merupakan contoh dari *selection*.

- **Mengubah mode heat dan cold:** Program memeriksa apakah pengguna ingin mengganti mode AC cold atau heat dengan menekan tombol mode pada UI.

```
if heat == True:  
    simbolheat.config(text="\u2600")  
    simbolhumidity.config(text="")  
elif cold == True:  
    simbolhumidity.config(text="\ud83d\udca7")  
    simbolheat.config(text="")
```

Ini adalah *selection* karena program memeriksa kondisi pilihan dan menyesuaikan nilai mode berdasarkan pilihan pengguna.

C. Repetisi (*Repetition*)

Repetisi adalah bagian dari kode yang akan diulang selama kondisi tertentu dipenuhi. Biasanya menggunakan *loop*.

Contoh dalam *code*:

- **Pengulangan code:** Bagian ini akan mengulang code sampai code valid/cocok.

```
for code in ir_codes:
    # Kirim kode IR
    sent_code = send_ir_code(code)
    # Simulasikan penerimaan dan respons AC
    if receive_ir_code(sent_code, ac_target_code):
        print(f"Remote berhasil disinkronkan dengan {sent_code}.\n")
        return code
    time.sleep(1) # Tunggu sebelum mencoba kode berikutnya
```

Tidak ada struktur repetisi eksplisit dalam kode ini, tetapi repetisi dapat muncul jika fungsi seperti `updateelayar()` dipanggil berulang kali.

D. Daftar (Array)

Array adalah struktur data yang menyimpan sejumlah nilai dalam satu variabel, biasanya dengan indeks untuk mengakses elemen tertentu.

Contoh dalam *code*:

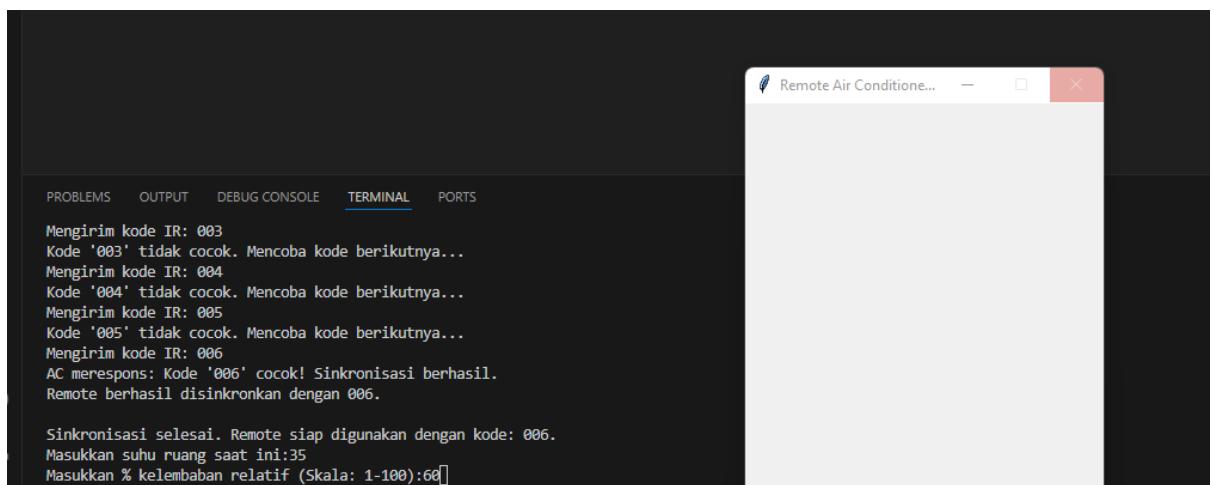
- Code IR: Array ini menyimpan daftar kode IR bawaan

```
ir_codes = {
    "001": "Tipe A",
    "002": "Tipe B",
    "003": "Tipe C",
    "004": "Tipe D",
    "005": "Tipe E",
    "006": "Tipe F",
    "007": "Tipe G",
    "008": "Tipe H",
    "009": "Tipe I",
    "010": "Tipe J",
}
```

Dengan menggunakan *array* ini, dapat dengan mudah mencocokkan merek remote AC (SIMULASI)

- **Mode:** Mode yang digunakan oleh AC disimpan dalam variabel mode, yang pada awalnya berisi "*Cooling*". Meskipun ini bukan *array* dalam bentuk struktur data yang lebih kompleks, namun terdapat banyak pilihan mode yang dapat dipilih, dan ini diperlakukan seperti *array* dengan pilihan yang bisa dipilih.

3.7. Implementasi Program



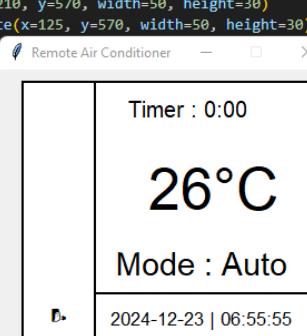
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Mengirim kode IR: 003
Kode '003' tidak cocok. Mencoba kode berikutnya...
Mengirim kode IR: 004
Kode '004' tidak cocok. Mencoba kode berikutnya...
Mengirim kode IR: 005
Kode '005' tidak cocok. Mencoba kode berikutnya...
Mengirim kode IR: 006
AC merespons: Kode '006' cocok! Sinkronisasi berhasil.
Remote berhasil disinkronkan dengan 006.

Sinkronisasi selesai. Remote siap digunakan dengan kode: 006.
Masukkan suhu ruang saat ini:35
Masukkan % kelembaban relatif (Skala: 1-100):60
```

1. Saat pertama kali *source code* di-*Run*, code akan mensimulasikan pengiriman kode IR selayaknya remote asli dan *user* diminta untuk memasukkan suhu ruangan saat ini/suhu ruangan dalam dan nilai *relative humidity* (kelembaban ruangan), dimana nantinya ini akan mempengaruhi status kompressor, *fan indoor*, dan *fan outdoor*.

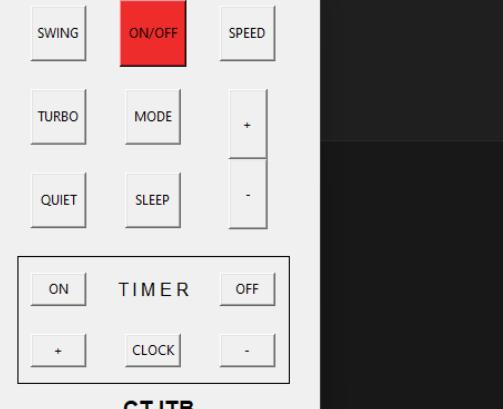
```
C:\Users\aldie\Downloads\Remote AC\final.py
477     Button(text="OFF", command=lambda: ubahnyalatimer("OFF")).place(x=210, y=515, width=50, height=30)
478     Button(text="-", command=lambda: ubahtimer(-1)).place(x=210, y=570, width=50, height=30)
479     Button(text="CLOCK", command=lambda: ubahnyalajam()).place(x=125, y=570, width=50, height=30)
480
481 # ----- MENAMPAKILAN JENDELA -----
482 window.mainloop()
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Kode '003' tidak cocok. Mencoba kode berikutnya...
Mengirim kode IR: 004
Kode '004' tidak cocok. Mencoba kode berikutnya...
Mengirim kode IR: 005
Kode '005' tidak cocok. Mencoba kode berikutnya...
Mengirim kode IR: 006
AC merespons: Kode '006' cocok! Sinkronisasi berhasil.
Remote berhasil disinkronkan dengan 006.

Sinkronisasi selesai. Remote siap digunakan dengan kode: 006.
Masukkan suhu ruang saat ini:35
Masukkan % kelembaban relatif (Skala: 1-100):60
```



2. Selanjutnya *user* dapat melihat tampilan remote di atas dan menekan tombol manapun untuk menjalankan perintah tertentu. Dengan adanya fitur memory,

apabila remote sempat diakses sebelumnya maka suhu yang keluar akan mengikuti setelan sebelumnya.

```
AC merespons. Kode 000 COCOK: SINKRONISASI DERNASII.
Remote berhasil disinkronkan dengan 006.

Sinkronisasi selesai. Remote siap digunakan dengan kode: 006
Masukkan suhu ruang saat ini:35
Masukkan % kelembaban relatif (Skala: 1-100):60
Status Swing: Menyala
```



3. Di sini kita coba perintah mulai dari yang ke-1: *swing*. Di gambar terlihat setelah tombol swing ditekan, maka akan muncul status swing menyala pada terminal dan muncul ikon swing pada tampilan *remote*.

```
454 |     print("Mode Sleep selesai. Suhu akhir:", suhu, "\n-----")
455 |
456 |
457 |
458 # ----- TOMBOL UTAMA BAGIAN TENGAH -----
459 Button(text="ON/OFF", command=onoff, background="firebrick2", activebackground="firebrick2", width=50, height=50)
460 Button(text="SWING", command=lambda: ubahswing()).place(x=40, y=275, width=50, height=50)
461 Button(text="SPEED", command=lambda: ubahkecepatan()).place(x=210, y=275, width=50, height=50)
462 Button(text="MODE", command=lambda: ubahmode()).place(x=125, y=350, width=50, height=50)

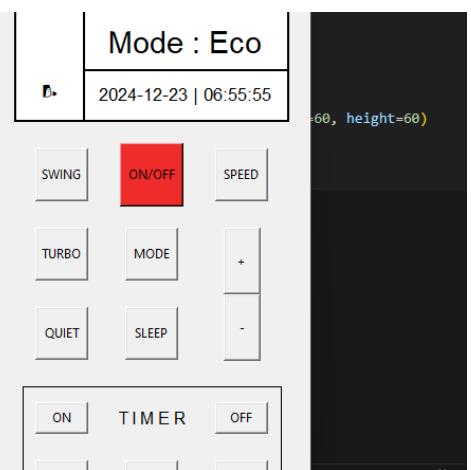
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

Mode Cool: Suhu ruang = 35.0, Suhu target = 26
Kompressor: 8/10, Fan Indoor: 7/10, Fan Outdoor: 7/10

Mode Dry: Humiditas = 60.0%
Kompressor: 6/10, Fan Indoor: 4/10, Fan Outdoor: 5/10

Mode Fan: Hanya kipas yang bekerja
Kompressor: 0/10, Fan Indoor: 5/10, Fan Outdoor: 0/10

Mode Eco: Suhu ruang = 35.0, Suhu target = 26
Kompressor: 5/10, Fan Indoor: 4/10, Fan Outdoor: 4/10



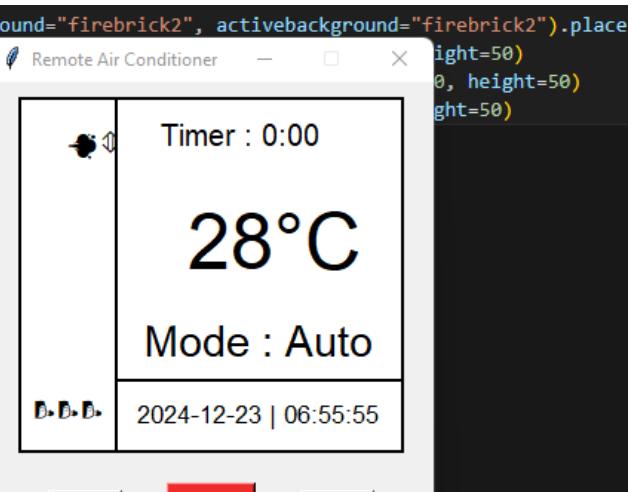
4. Di sini contoh meng-*input* menu 2 yaitu mode. Dalam mode ini terdapat beberapa mode yaitu *auto* sebagai mode default AC, ada mode fan, cool, dry, serta dua mode yang memiliki button sendiri yakni turbo and quiet. Untuk mode yang berada pada tombol ‘MODE’ hanya berpengaruh pada kerja kompressor, fan indoor, dan outdoor selayaknya yang terdapat pada terminal. Adapun tombol turbo and quiet akan memunculkan ikon ‘tiup’ pada kiri atas.

```
459 Button(text="ON/OFF", command=onoff, background="firebrick2", activebackground="firebrick2").place(x=40, y=275, width=50, height=50)
460 Button(text="SWING", command=lambda: ubahswing()).place(x=40, y=275, width=50, height=50)
461 Button(text="SPEED", command=lambda: ubahkecepatan()).place(x=210, y=275, width=50, height=50)
462 Button(text="MODE", command=lambda: ubahmode()).place(x=125, y=350, width=50, height=50)

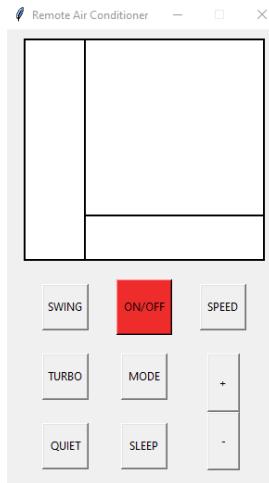
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

Mode Auto: Suhu ruang = 35.0, Suhu target = 26
Kompressor: 10/10, Fan Indoor: 10/10, Fan Outdoor: 10/10

Mode Turbo: Fan Indoor bekerja maksimal (Fan Indoor: 10/10)
Suhu diinginkan saat ini: 27
Suhu diinginkan saat ini: 28
Suhu diinginkan saat ini: 27
Suhu diinginkan saat ini: 26
Suhu diinginkan saat ini: 25
Suhu diinginkan saat ini: 26
Suhu diinginkan saat ini: 27
Suhu diinginkan saat ini: 28



5. Menekan tombol (+/-) pada remote akan menaikkan/ menurunkan suhu target selayaknya bagaimana remote biasa bekerja. Nilai suhu terbaru akan juga ditampilkan pada terminal.



6. Selanjutnya menu terakhir, yaitu *Off AC*, *user* menekan tombol on/off untuk mematikan AC. Dalam status bar *remote*, status menghilang semua yang menandakan bahwa AC telah mati.

BAB 4

KESIMPULAN & LESSON LEARNED

4.1 Kesimpulan

- Penerapan *Computational Thinking*

Melalui pendekatan *Computational Thinking* (CT), sistem kerja AC yang kompleks dapat disederhanakan menjadi sub-sistem yang lebih mudah dipahami. Pendekatan ini memanfaatkan dekomposisi, abstraksi, pengenalan pola, dan algoritma untuk menciptakan solusi yang efektif dan efisien.

- Pemahaman Sistem AC

Sistem AC dikaji dan dirancang ulang berdasarkan fungsi-fungsi utama seperti pengaturan suhu, mode operasi, dan kecepatan kipas. Komponen-komponen utama seperti evaporator, kompresor, kondensor, katup ekspansi, dan kipas diidentifikasi perannya untuk menyusun sistem yang lebih terstruktur.

- Algoritma yang Sederhana dan Terstruktur

Algoritma untuk mengontrol fungsi dasar AC berhasil dirancang menggunakan konsep seperti sekuens, seleksi, repetisi, dan array. Algoritma ini memodelkan mekanisme kerja AC yang realistik namun tetap sederhana.

- Implementasi Program

Program simulasi AC yang dibuat mampu merepresentasikan fungsi-fungsi utama sistem AC, seperti pengaturan suhu, mode, kecepatan kipas, dan fitur *swing*. Simulasi ini menampilkan interaksi pengguna dengan *remote AC* melalui *input* dan *output* yang realistik, meningkatkan pemahaman tentang kerja AC.

- Pengembangan Keterampilan

Proyek ini membantu mengembangkan keterampilan dalam *Computational Thinking*, pemrograman, dan pemahaman sistem teknis yang kompleks, memberikan bekal untuk penerapan konsep serupa dalam berbagai teknologi lain.

4.2 Lesson learned

Proyek ini menunjukkan bagaimana pendekatan CT dapat digunakan secara efektif untuk menyederhanakan, memahami, dan memodelkan sistem teknis kompleks, serta mengimplementasikannya ke dalam sebuah program simulasi. Kami belajar untuk melakukan dekomposisi terhadap masalah yang ada dan mengimplementasikannya ke dalam sebuah program.

Link Video Presentasi dan Source Code:

<https://drive.google.com/file/d/1-MS3bfVTFETwzH-OpGMp0-GWJhwzrJXm/view?usp=sharing>

DAFTAR PUSTAKA

- Tim Penyusun Materi WI1102. (2024). **Algorithmic Thinking**. Institut Teknologi Bandung.
- Bebras Indonesia. (2024). **Bebras Indonesia Book 2020 - SMA (Edisi Oktober 2024)**. Diakses dari <https://bebras.or.id/v3/wp-content/uploads/2024/10/Bebras-Indonesia-Book-2020-SMA-OK-Okt2024.pdf>
- As'ari, M. A. (2016). **SPLIT – AIR CONDITIONER**. Diakses dari <https://www.google.com/amp/s/slidesplayer.info/amp/12313625/>
- Super Radiator Coils. (2021). **The 4 Main Refrigeration Cycle Components**. Diakses dari <https://www.superradiatorcoils.com/blog/4-main-refrigeration-cycle-components>
- Tarigan, G. K. (2021). **Mengenal Siklus Carnot & Penerapannya Dalam Kehidupan Sehari-hari**. Diakses dari <https://www.indonesiana.id/read/150427/mengenal-sehari-hari>
- Quality Technic. (2015). *Pengertian Dasar Tentang AC (Air Conditioner)*. Diakses dari <https://www.blog.qualitytechnic.com/2015/04/pengertian-dasar-tentang-ac-air-conditioner.html?m=1>

PEMBAGIAN KERJA

Pembagian tugas utama tiap anggota:

Note: ini adalah pembagian tugas **utama**, yang berarti tiap anggota kelompok bertanggung jawab atas tugas utamanya dan tiap anggota tetap saling melengkapi, saling mengoreksi, dan saling membantu satu sama lain.

No.	Pekerjaan	Anggota Kelompok	NIM
1.	Flowchart, Penambahan Fitur Mode	Rhenaldy Cahyadi Putra	19624019
2.	Penambahan Simulasi Remote IR	Suryani Mulia Utami	19624020

3.	Penambahan GUI	Muhammad Haris Putra S.	19624032
4.	Penambahan Timer & Memory	Miguel Rangga Deardo Sinaga	19624053
5.	Revisi, Finalisasi Laporan Akhir	Muchammad Rafif Azis Syahlevi	19624061