

TUGAS TEORI BAHASA DAN OTOMATA
PROBLEM SOLVING KASUS *VENDING MACHINE* PADA MESIN
PENJUALAN BUSANA ADAT OTOMATIS DI PURA TAMPAK SIRING



Dosen Pengampu:

Dr. Anak Agung Istri Ngurah Eka Karyawati, S.Si., M.Eng.

Oleh Kelompok: C5

- | | |
|-----------------------------|--------------|
| 1. Ni Putu Dita Ari Pratiwi | (2208561003) |
| 2. Putu Ardi Sudarmika | (2208561028) |
| 3. Putu Yuki Parmawati | (2208561066) |
| 4. Intara Pratama Harahap | (2208561104) |

PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA

2023

KATA PENGANTAR

Puji syukur kami panjatkan kehadapan Tuhan Yang Maha Esa atas karunianya kami dapat menyelesaikan makalah ini guna memenuhi tugas kelompok untuk mata kuliah Teori Bahasa dan Otomata dengan judul “*Problem Solving Kasus Vending Machine* Pada Mesin Penjualan Busana Adat Otomatis Di Pura Tampak Siring”.

Kami menyadari bahwa dalam penulisan laporan ini tidak terlepas dari bantuan banyak pihak yang dengan tulus memberikan doa saran dan kritik sehingga dapat terselesaikan dengan baik.

Kami menyadari sepenuhnya bahwa makalah ini jauh dari kata sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang kami miliki. Oleh karena itu kami mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari pembaca sekalian. Akhirnya kami berharap bahwa laporan ini dapat memberikan manfaat kepada pembaca.

Jimbaran, 21 Oktober 2023

Tim Penyusun

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Merumuskan Masalah.....	1
1.3 Tujuan.....	2
BAB II PERMASALAHAN.....	3
2.1 Identifikasi Permasalahan.....	3
2.2 Batasan.....	3
2.3 Asumsi.....	4
BAB III PENYELESAIAN.....	5
3.1 Finite State Automata.....	5
3.2 Perancangan Sistem Vending Machine.....	11
3.3 Implementasi Algoritma Finite State Automata Program Python.....	13
3.4 Pengujian Vending Machine.....	32
BAB IV MANUAL APLIKASI.....	34
4.1 Fitur Sistem.....	34
4.2 Antarmuka.....	34
BAB V PENUTUP.....	42
5.1 Kesimpulan.....	42

BAB I

PENDAHULUAN

1.1 Latar Belakang

Bali terkenal dengan julukannya sebagai pulau seribu pura, potensi wisata yang besar menjadikannya sebagai objek wisata pura dengan keunikan budaya, seni yang indah, dan pemandangan yang menakjubkan. Penting untuk selalu menghormati dan memahami adat dan tradisi setempat saat mengunjungi pura agama Hindu di Bali, salah satu aspek dalam etika memasuki area pura Hindu di Bali adalah menggunakan busana adat yang benar. Namun, dalam kenyataan di lapangan dapat ditemukannya, para wisatawan domestik maupun mancanegara yang menggunakan pakaian minim atau tidak sopan di tempat yang tidak seharusnya seperti pada tahun 2016, seorang wanita difoto dengan pose yoga serta menggunakan bikini di depan pura, himbauan untuk menggunakan pakaian yang sopan saat masuk ke tempat suci juga dikonfirmasi oleh Kepala Dinas Pariwisata Bali (dikutip dari kumparan.com).

Sebagai bentuk salah satu solusi masalah tersebut, kami memperkenalkan konsep mesin penjual busana adat otomatis yang diletakkan pada objek wisata pura, yang dapat memberikan kemudahan bagi wisatawan serta sebagai bentuk inovasi teknologi dalam mendukung kenyamanan wisatawan. Dengan harapan mesin ini mampu memberikan kemudahan ketika pengunjung mencari busana adat terdekat sebelum masuk ke area tempat suci, bagi wisatawan yang kurang mempersiapkan pakaian adat contohnya saat melukat di pura Tampak Siring karena kunjungan yang kurang terencana, serta keberadaan mesin ini dapat beroperasi 24 jam sehingga mengurangi ketergantungan pada sumber daya manusia.

Oleh karena itu, solusi inovatif *vending machine* dengan konsep *Finite State Automata* (FSA) menjadi alternatif yang dapat membantu mengatasi sejumlah masalah tersebut dan membawa pengalaman pengunjung ke tingkat yang lebih baik serta tetap untuk menghormati dan memahami adat setempat. Dengan menggabungkan teknologi dan model matematika FSA, diharapkan *vending machine* ini dapat memberikan solusi yang komprehensif dan efisien.

1.2 Merumuskan Masalah

Rumusan masalah dari laporan kami adalah :

1. Apa identifikasi permasalahan yang diangkat?

2. Bagaimana cara mengimplementasikan *finite state automata* serta mengevaluasi penyelesaian kasus pada mesin Penjualan Busana Adat Otomatis?
3. Bagaimana cara membangun skenario eksperimen berupa perancangan sistem yang digunakan serta menentukan ukuran evaluasi yang digunakan?
4. Bagaimana metode dan algoritma *finite state automata* dapat diimplementasikan pada mesin penjualan busana adat otomatis?
5. Bagaimana desain perancangan *vending machine* dan pengujian validasi kebergunaan sistem?

1.3 Tujuan

Adapun tujuan dari perancangan laporan kami adalah sebagai berikut :

1. Dapat memberikan solusi dari identifikasi permasalahan terkait mesin penjualan busana adat otomatis
2. Mahasiswa diharapkan mampu melakukan perancangan finite state automata sebagai bentuk evaluasi penyelesaian kasus pada mesin penjualan busana adat otomatis
3. Mahasiswa diharapkan mampu dalam membangun skenario eksperimen berupa perancangan sistem serta menentukan evaluasi yang digunakan berdasarkan identifikasi permasalahan pada mesin penjualan busana adat otomatis
4. Mahasiswa dapat menerapkan metode dan algoritma *finite state automata* yang diimplementasikan ke dalam mesin penjualan busana adat otomatis
5. Mahasiswa dapat melakukan perancangan desain *vending machine* dan melakukan pengujian validasi kebergunaan sistem

BAB II

PERMASALAHAN

2.1 Identifikasi Permasalahan

Terdapat dua permasalahan utama yang kami angkat dalam laporan ini terkait perancangan *Vending Machine* untuk penjualan busana adat otomatis untuk objek wisata pura di Bali yang memiliki banyak pengunjung, baik domestik maupun mancanegara, seperti Tampak Siring. Pertama ialah tidak memadainya persiapan wisatawan. Terdapat beberapa kejadian dimana dijumpai wisatawan yang menggunakan pakaian yang kurang sopan masuk ke tempat suci, wisatawan atau pemedek yang berkunjung tanpa perencanaan sebelumnya sehingga tidak menggunakan atau membawa busana adat yang sepantasnya ketika memasuki area pura, dan pengunjung yang ingin melukat di pura namun tidak membawa busana adat. Kedua ialah ketergantungan pada tenaga manusia membuat akses terhadap busana adat terbatas dalam konteks waktu. Penjualan busana adat yang dioperasikan oleh manusia tidak memungkinkan bisa diaksesnya busana adat tersebut dalam 24/7. Wisatawan tidak akan bisa membeli ataupun menyewa busana adat di luar jam operasi.

Sehingga tujuan perancangan *vending machine* menggunakan konsep *finite state automata* (FSA) untuk penjualan busana adat otomatis ini adalah mengatasi permasalahan tersebut. dengan membuat mesin yang menyediakan busana adat didalamnya serta pengguna dapat melakukan pembelian. Sedangkan penerapan konsep FSA terdapat pada proses pembelian dan pembayaran busana adat.

2.2 Batasan

Berikut ini merupakan beberapa batasan dari *vending machine* yang kami rancang:

1. Keterbatasan pilihan produk yang ditawarkan, dimana pada mesin ini hanya menyediakan busana adat penting atau yang wajib ada untuk memasuki area tempat suci.
2. Keterbatasan pada pilihan ukuran busana. dimana pada mesin ini ukuran busana yang disediakan untuk wanita atau pria yaitu M, L, XL
3. Keterbatasan metode pembayaran, yang dimana *user* tidak bisa menggunakan metode lain selain Qris dan *e-money*. Contohnya tunai tidak bisa digunakan.
4. Keterbatasan pada target *user*, dimana tidak semua kalangan bisa menggunakan. Contoh anak-anak dan lansia.
5. Keterbatasan pada stok produk. Hanya ada 20 buah di setiap mesin.

6. Keterbatasan pada pilihan model busana, dimana tidak semua model busana adat tersedia di mesin.

2.3 Asumsi

Berikut ini merupakan beberapa asumsi yang dapat dipertimbangkan dalam perancangan *vending machine* ini:

1. *User* mampu mengoperasikan *vending machine*. Dalam asumsi dasar, kami berasumsi bahwa *user* memiliki kemampuan dasar untuk mengoperasikan *vending machine*. Mereka harus dapat menggunakan layar sentuh atau antarmuka yang disediakan dengan mudah. Pada rancangan *vending machine* ini juga akan menampilkan peraturan penggunaan mesin sehingga dapat memudahkan *user*
2. *User* mampu serta memahami cara melakukan pembayaran pada *vending machine* yang menerima bentuk pembayaran berupa *e-money* melalui *barcode* yang tersedia.
3. *User* dapat atau mampu memilih opsi sesuai yang telah ditampilkan di layar terkait busana adat wanita atau pria
4. Mesin telah memiliki 20 stok untuk setiap itemnya, dengan asumsi barang dimasukkan secara manual kedalam mesin, serta stok barang akan kembali 20 apabila admin atau pemilik mesin melakukan pengisian.
5. *User* mengetahui cara pengambilan busana yang berada di bagian bawah mesin.
6. Busana yang dikeluarkan benar tanpa adanya rusak produk
7. *Vending machine* tersedia dan berfungsi dengan baik. Kami berasumsi bahwa *vending machine* selalu tersedia dan dalam kondisi baik. Dalam kasus pemeliharaan atau masalah teknis, ada rencana darurat yang tersedia.

BAB III PENYELESAIAN

3.1 Finite State Automata

Pada tahap ini seluruh alur proses yang telah ditentukan akan dituangkan ke dalam serangkaian diagram *state*. Diagram *finite state automata* ini menggunakan jenis FSA non-deterministik (NFA) dengan mempertimbangkan cara kerja dan karakteristik simulasi mesin yang dirancang. Perancangan diagram FSA dapat dilihat pada gambar berikut:

Tabel 1. Tipe Produk Dan Harga

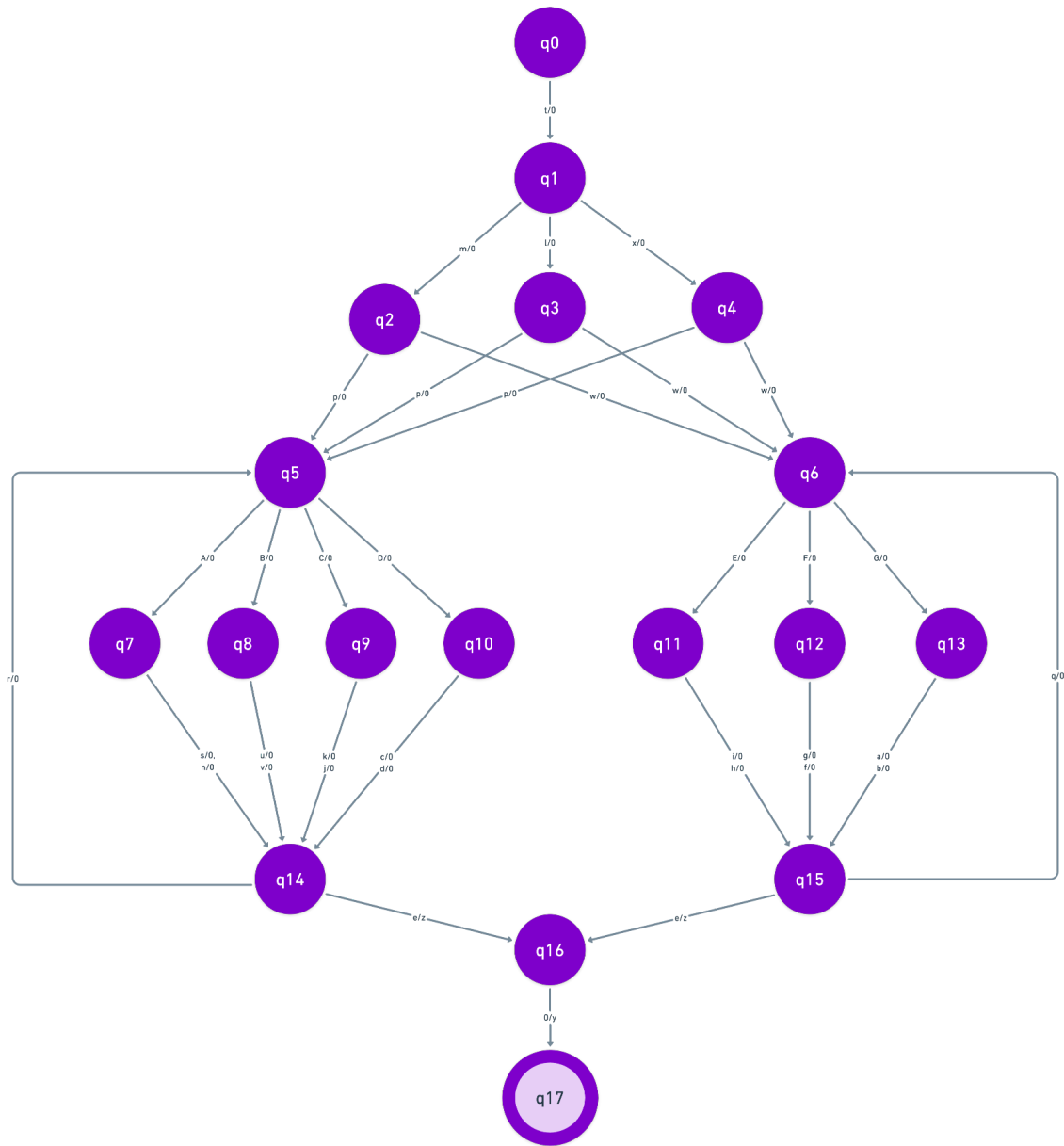
No	Gender	Tipe Produk	Ukuran	Desain	Harga
1	pria	udeng	M, L, XL	belum jadi	Rp40.000
2				sudah jadi	Rp55.000
3		baju adat bali	M, L, XL	safari	Rp100.000
4				kemeja	Rp90.000
5				Lembaran	Rp80.000
6		kamben dan saputan	M, L, XL	setengah jadi	Rp85.000
8		selendang pria	M, L, XL	bermotif	Rp25.000
9				polos	Rp15.000
10	wanita	kebaya	M, L, XL	kain brokat	Rp100.000
11				kain sari	Rp90.000
12		kamben	M, L, XL	belum jadi	Rp110.000
13				sudah jadi	Rp150.000
14				bermotif	Rp25.000
15		selendang wanita	M, L, XL	tanpa motif	Rp15.000

3.1.1 Membangun Diagram Transisi

Tabel 2. Identifikasi *input/output* dan *state*

Kode	Uraian	Ket	Kode	Uraian	Ket
q0	State awal	State	y	Mengeluarkan produk	Output
q1	Menyiapkan tas	State	t	Memilih mulai	Input
q2	Ukuran M	State	m	Memilih ukuran M	Input
q3	Ukuran L	State	l	Memilih ukuran L	Input
q4	Ukuran XL	State	x	Memilih ukuran XL	Input
q5	Menampilkan pilihan pria	State	p	Memilih pria	Input

q6	Menampilkan pilihan wanita	State	w	Memilih wanita	Input
q7	Baju adat pria	State	s	Memilih baju safari	Input
q8	Udeng	State	n	Memilih baju kemeja	Input
q9	Kamen dan saput	State	u	Memilih udeng lembaran	Input
q10	Selendang pria	State	v	Memilih udeng jadi	Input
q11	Kebaya wanita	State	k	Memilih kamen dan saput lembaran	Input
q12	Kamen wanita	State	j	Memilih kamen dan saput setengah jadi	Input
q13	Selendang wanita	State	c	Memilih selendang pria polos	Input
q14	Konfirmasi selesai belanja pada pria	State	d	Memilih selendang pria motif	Input
q15	Konfirmasi selesai belanja pada wanita	State	i	Memilih kebaya brokat	Input
q16	mengeluarkan Qris	State	h	Memilih kebaya sari	Input
q17	Produk keluar	State	g	Memilih kamen lembaran	Input
E	Memilih jenis kebaya wanita	Input	f	Memilih kamen jadi	Input
F	Memilih jenis kamen wanita	Input	a	Memilih selendang wanita polos	Input
G	Memilih jenis selendang wanita	Input	b	Memilih selendang wanita motif	Input
r	Memilih konfirmasi lanjut pria	Input	A	Memilih jenis baju adat pria	Input
q	memilih konfirmasi lanjut wanita	Input	B	Memilih jenis udeng	Input
e	Memilih konfirmasi selesai	Input	C	Memilih jenis kamen dan saput	Input
z	Mengeluarkan Qris	Output	D	Memilih jenis selendang pria	Input



Gambar 1. Rancangan Diagram *Finite State*

Gambar tersebut memvisualisasikan setiap keadaan (state) yang mungkin terjadi dalam sistem, bersama dengan transisi-transisi antara keadaan tersebut. Dengan menggunakan pendekatan NFA, sistem dapat memiliki lebih dari satu transisi yang mungkin untuk keadaan dan input tertentu, memberikan fleksibilitas dalam perancangan dan simulasi. Adapun untuk pendefinisian tuplenya adalah sebagai berikut :

Pendefinisian Tuple

FSA *Output* didefinisikan sebagai berikut:

$$M = (Q, \Sigma, S, \delta, \Delta, \lambda)$$

Yang mana:

Q = himpunan *state*, Σ = simbol *input*, δ = fungsi transisi, S = *state* awal, Δ = himpunan simbol *output*, λ = fungsi *output*.

Sehingga dapat didefinisikan sebagai berikut:

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, q_{17}\}$$

$$\Sigma = \{y, t, m, l, x, p, w, s, n, u, v, k, j, c, d, i, h, g, f, a, b, A, B, C, D, E, F, G, r, q, e, z\}$$

δ = fungsi transisi

$$\begin{aligned} \delta(q_0, t) &= q_1; & \delta(q_1, m) &= q_2; & \delta(q_1, l) &= q_3; & \delta(q_1, x) &= q_4; & \delta(q_2, p) &= q_5; \\ \delta(q_2, w) &= q_6; & \delta(q_3, p) &= q_5; & \delta(q_3, w) &= q_6; & \delta(q_4, p) &= q_5; & \delta(q_4, w) &= q_6; \\ \delta(q_2, p) &= q_5; & \delta(q_5, A) &= q_7; & \delta(q_5, B) &= q_8; & \delta(q_5, C) &= q_9; & \delta(q_5, D) &= q_{10}; \\ \delta(q_6, E) &= q_{11}; & \delta(q_6, F) &= q_{12}; & \delta(q_6, G) &= q_{13}; & \delta(q_7, s) &= q_{14}; & \delta(q_7, n) &= q_{14}; \\ \delta(q_8, u) &= q_{14}; & \delta(q_8, v) &= q_{14}; & \delta(q_9, k) &= q_{14}; & \delta(q_9, j) &= q_{14}; & \delta(q_{10}, c) &= q_{14}; \\ \delta(q_{10}, d) &= q_{14}; & \delta(q_{11}, i) &= q_{15}; & \delta(q_{11}, h) &= q_{15}; & \delta(q_{12}, g) &= q_{15}; & \delta(q_{12}, f) &= q_{15}; \\ \delta(q_{13}, b) &= q_{15}; & \delta(q_{13}, a) &= q_{15}; & \delta(q_{14}, r) &= q_5; & \delta(q_{14}, e) &= q_{16}; & \delta(q_{15}, q) &= q_6; \\ \delta(q_{15}, e) &= q_{16}; \end{aligned}$$

$$S = \{q_0\}$$

$$\Delta = \{0, z, y\}$$

λ = fungsi *output* untuk setiap transisi

$$\begin{aligned} \lambda(q_0, t) &= 0; & \lambda(q_1, m) &= q_2; & \lambda(q_1, l) &= q_3; & \lambda(q_1, x) &= q_4; & \lambda(q_2, p) &= q_5; \\ \lambda(q_2, w) &= q_6; & \lambda(q_3, p) &= q_5; & \lambda(q_3, w) &= q_6; & \lambda(q_4, p) &= q_5; & \lambda(q_4, w) &= q_6; \\ \lambda(q_2, p) &= q_5; & \lambda(q_5, A) &= q_7; & \lambda(q_5, B) &= q_8; & \lambda(q_5, C) &= q_9; & \lambda(q_5, D) &= q_{10}; \\ \lambda(q_6, E) &= q_{11}; & \lambda(q_6, F) &= q_{12}; & \lambda(q_6, G) &= q_{13}; & \lambda(q_7, s) &= q_{14}; & \lambda(q_7, n) &= q_{14}; \\ \lambda(q_8, u) &= q_{14}; & \lambda(q_8, v) &= q_{14}; & \lambda(q_9, k) &= q_{14}; & \lambda(q_9, j) &= q_{14}; & \lambda(q_{10}, c) &= q_{14}; \\ \lambda(q_{10}, d) &= q_{14}; & \lambda(q_{11}, i) &= q_{15}; & \lambda(q_{11}, h) &= q_{15}; & \lambda(q_{12}, g) &= q_{15}; & \lambda(q_{12}, f) &= q_{15}; \\ \lambda(q_{13}, b) &= q_{15}; & \lambda(q_{13}, a) &= q_{15}; & \lambda(q_{14}, r) &= q_5; & \lambda(q_{14}, e) &= z; & \lambda(q_{15}, q) &= q_6; \\ \lambda(q_{15}, e) &= z; & \lambda(q_{16}, 0) &= y; \end{aligned}$$

Tabel 3. Fungsi transisi *input*

	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17
y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
t	q1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

m	-	q2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
l	-	q3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
x	-	q4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
p	-	-	q5	q5	q5	-	-	-	-	-	-	-	-	-	-	-	-	-
w	-	-	q6	q6	q6	-	-	-	-	-	-	-	-	-	-	-	-	-
s	-	-	-	-	-	-	-	q14	-	-	-	-	-	-	-	-	-	-
n	-	-	-	-	-	-	-	q14	-	-	-	-	-	-	-	-	-	-
u	-	-	-	-	-	-	-	-	q14	-	-	-	-	-	-	-	-	-
v	-	-	-	-	-	-	-	-	q14	-	-	-	-	-	-	-	-	-
k	-	-	-	-	-	-	-	-	-	q14	-	-	-	-	-	-	-	-
j	-	-	-	-	-	-	-	-	-	q14	-	-	-	-	-	-	-	-
c	-	-	-	-	-	-	-	-	-	-	q14	-	-	-	-	-	-	-
d	-	-	-	-	-	-	-	-	-	-	q14	-	-	-	-	-	-	-
i	-	-	-	-	-	-	-	-	-	-	-	q15	-	-	-	-	-	-
h	-	-	-	-	-	-	-	-	-	-	-	q15	-	-	-	-	-	-
g	-	-	-	-	-	-	-	-	-	-	-	-	q15	-	-	-	-	-
f	-	-	-	-	-	-	-	-	-	-	-	-	q15	-	-	-	-	-
a	-	-	-	-	-	-	-	-	-	-	-	-	-	q15	-	-	-	-
b	-	-	-	-	-	-	-	-	-	-	-	-	-	q15	-	-	-	-
A	-	-	-	-	-	q7	-	-	-	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	q8	-	-	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	q9	-	-	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	q10	-	-	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	q11	-	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	q12	-	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	q13	-	-	-	-	-	-	-	-	-	-	-
r	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q5	-	-	-
q	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q6	-	-
e	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q16	q16	-	-
z	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	q17	-
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----	---

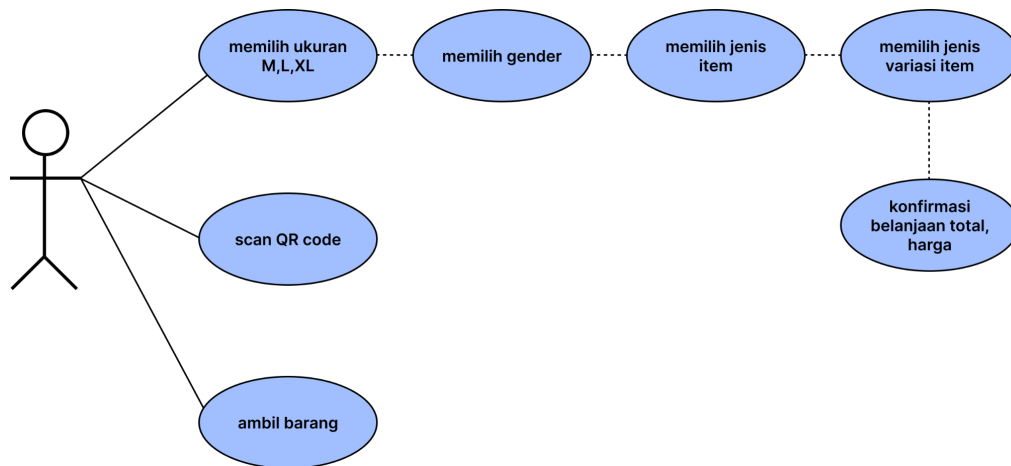
Tabel 4. Fungsi transisi *output*

	q0	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	q17
y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
t	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
m	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
l	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
x	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
p	-	-	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-
w	-	-	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-
s	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-
n	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-
u	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-
v	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-
k	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-
j	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-
c	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-
d	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-
i	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-
h	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-
g	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-
f	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-
a	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-
b	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-
A	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-

E	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-
r	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-
q	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-
e	-	-	-	-	-	-	-	-	-	-	-	-	-	-	z	z	-	-
z	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	y	-

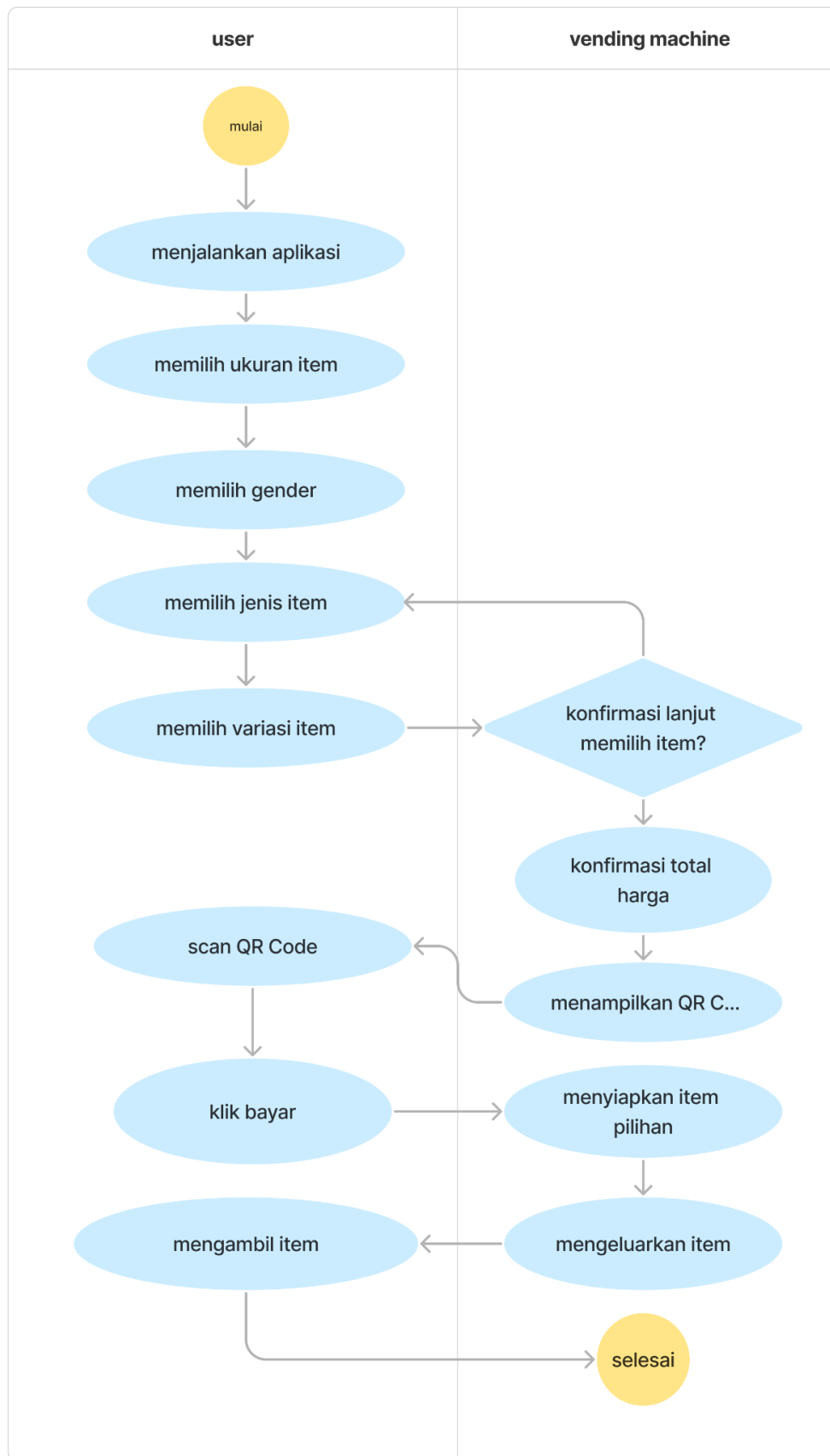
3.2 Perancangan Sistem Vending Machine

Dengan tujuan untuk memenuhi kebutuhan para pemakai (*user*) dan memberikan gambaran yang jelas tentang rancang bangun secara lengkap kepada programmer serta pihak yang terlibat. Adapun rancang bangun sistem informasi ini lebih banyak dipaparkan secara visual yang dilakukan kedalam beberapa bagian yaitu sebagai berikut. Langkah awalnya adalah membuat diagram kasus pengguna (*use case diagram*). *Use Case Diagram* adalah satu jenis dari diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor, perancangan *flowchart* untuk memudahkan dalam proses implementasi *code*, dan *activity diagram*



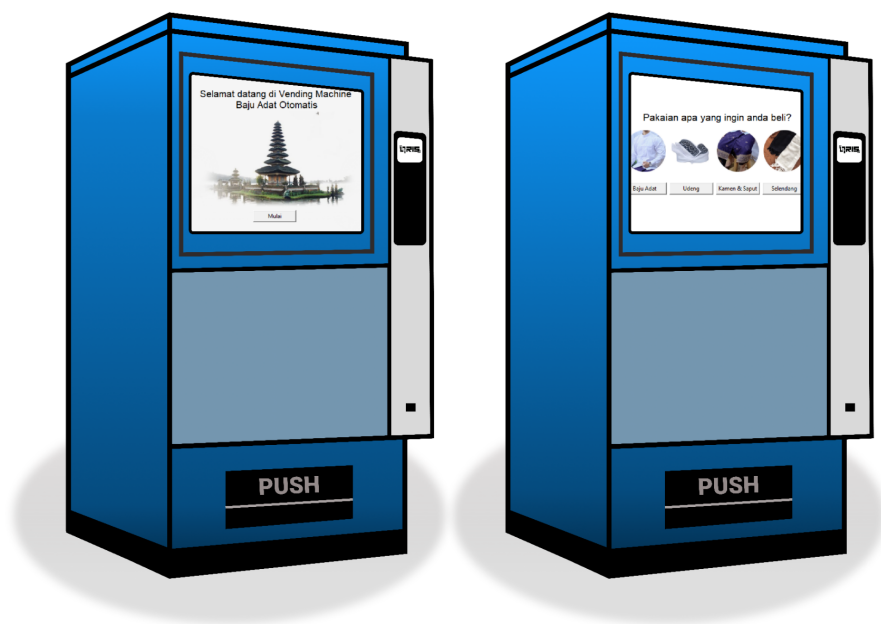
Gambar 2. *Use Case Diagram*

Dalam gambar tersebut merepresentasikan user dalam menggunakan *vending machine* penjual baju kebaya otomatis, dengan *use case* yang ada seperti pada gambar serta *use case* spesifikasinya dengan digambarkan pada garis putus - putus tersebut.



Gambar 3. *Activity Diagram*

Gambar *Activity Diagram* ini menggambarkan alur proses dalam menjalankan aplikasi. Alur dimulai ketika *user* menjalankan aplikasi, kemudian diminta untuk memilih ukuran item yang diinginkan, diikuti dengan pemilihan gender yang mungkin mempengaruhi penawaran item. Selanjutnya, *user* dapat memilih jenis item dan variasi item. Setelah memilih item, mereka diminta untuk mengkonfirmasi total harga sebelum melanjutkan ke tahap pembayaran. Dalam tahap pembayaran, *user* harus memberikan informasi pembayaran yang diperlukan, dan jika transaksi berhasil, sistem akan mengeluarkan *QR Code* sebagai bukti transaksi. *User* memiliki opsi untuk memindai *QR Code*, dan jika berhasil, transaksi dianggap berhasil. Setelah itu, *user* akan siap untuk mengambil item yang telah mereka beli. Jika transaksi tidak berhasil atau *user* memutuskan untuk tidak melanjutkan pembelian, mereka dapat kembali ke tahap menjalankan aplikasi dan memulai proses dari awal.



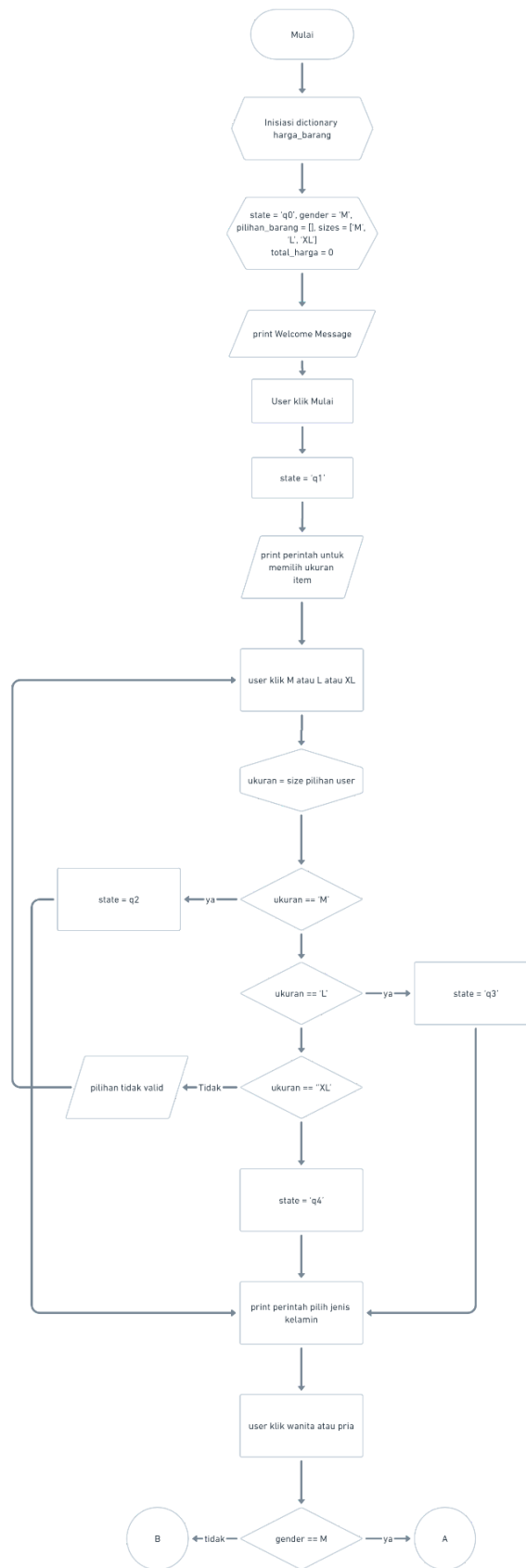
Gambar 4. Desain Mockup

Di atas ini merupakan desain *mockup vending machine* tampilan depan serta struktur mesinnya sendiri.

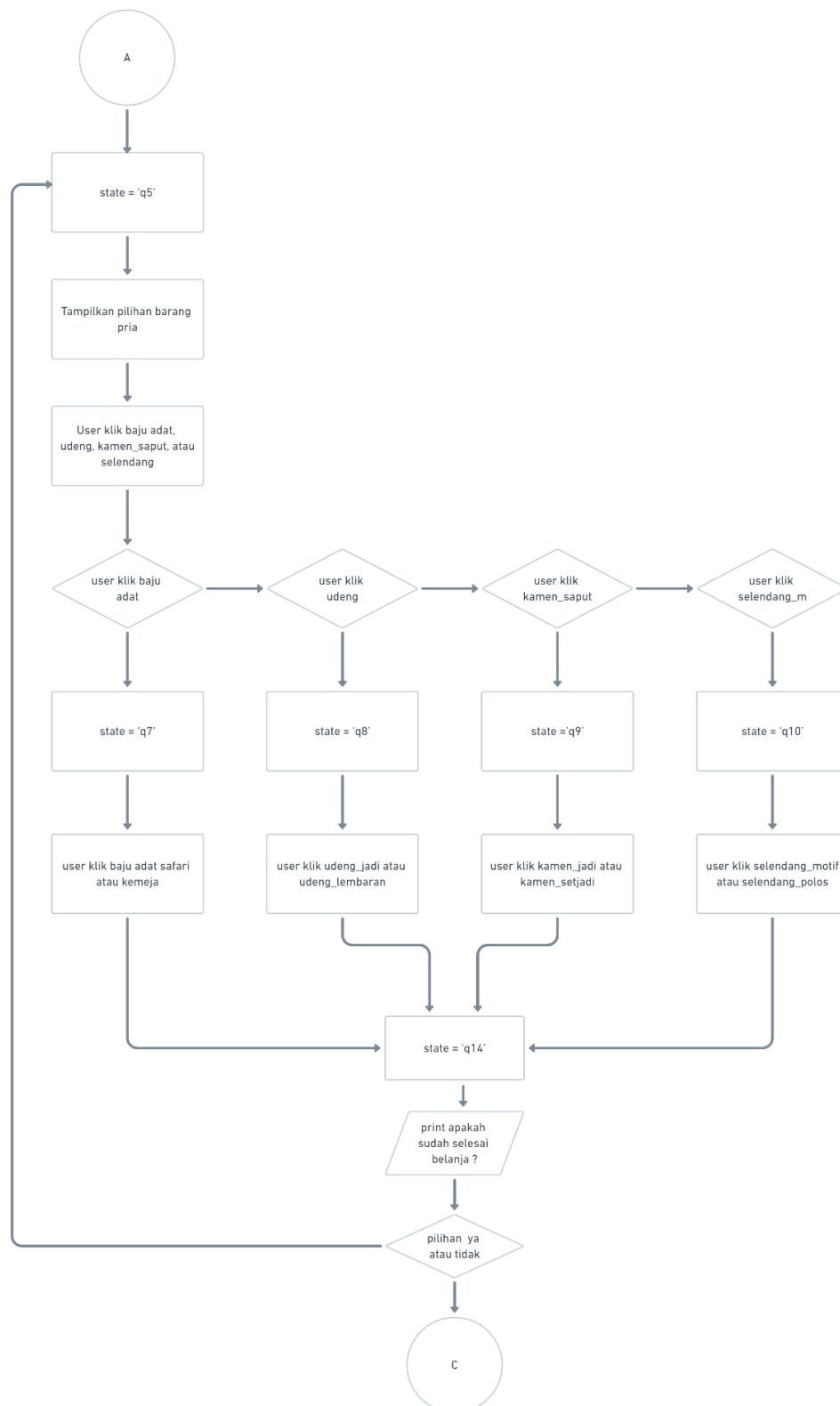
3.3 Implementasi Algoritma *Finite State Automata* Program Python

Implementasi dari algoritma *finite state automata* menggunakan bahasa pemrograman python

3.3.1 Flowchart



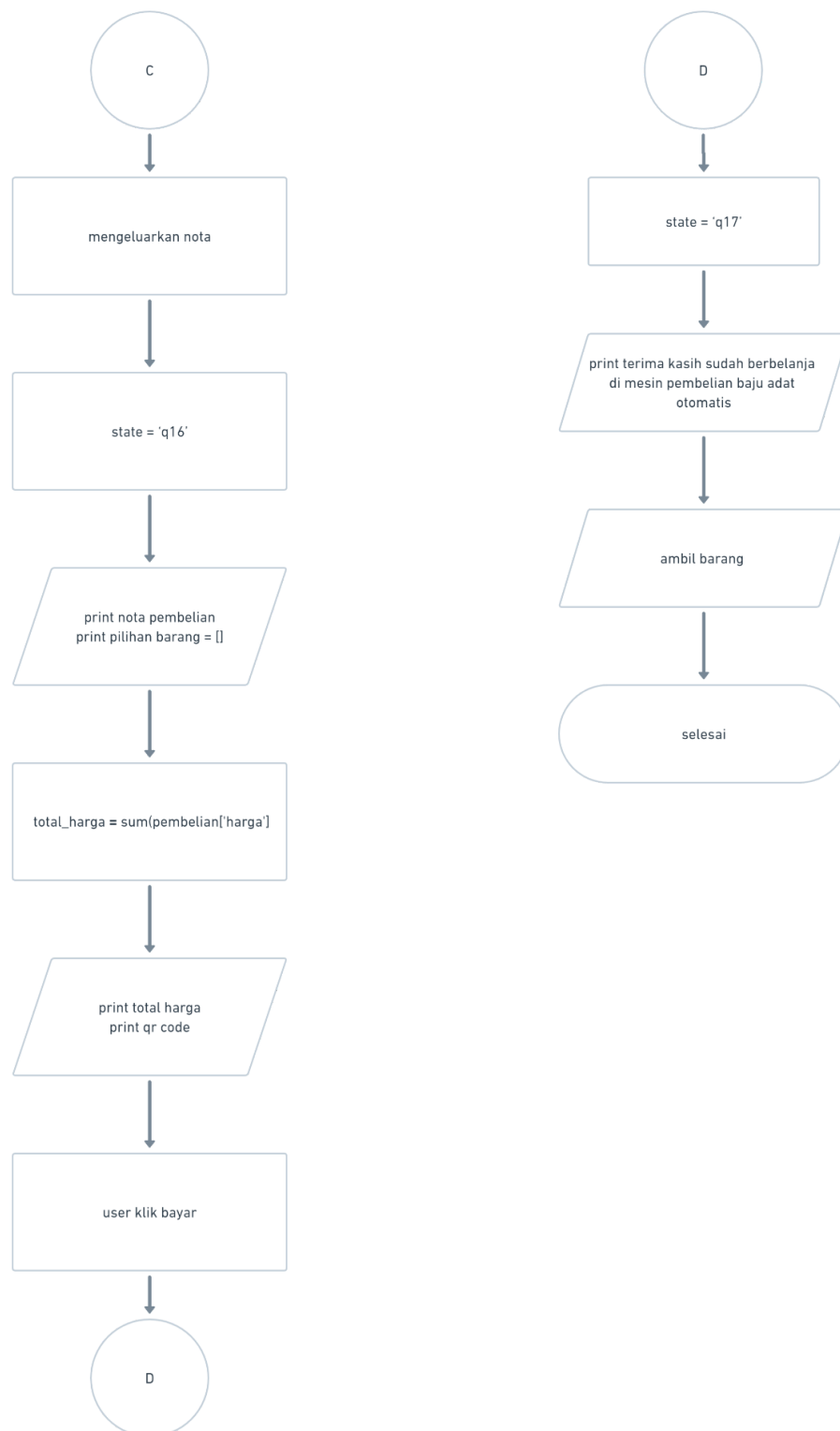
Gambar 5. *Flowchart* Menu Utama



Gambar 6. *Flowchart Menu Pria*



Gambar 7. *Flowchart Menu Wanita*



Gambar 8. *Flowchart* Nota dan Pembayaran

3.3.2 Program *Vending Machine*

```
import tkinter as tk
from PIL import Image, ImageTk

class GUI(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Vending Machine Baju Adat Otomatis")
        self.geometry('500x350')
        self.configure(bg="white")
        self.resizable(False, False)

        self.show_welcome_screen()

        self.harga_barang = {
            'Pria': {
                'Baju Adat': {'Safari': 100000, 'Kemeja': 90000},
                'Kamen': {'Lembaran': 80000, 'Setengah Jadi': 85000},
                'Udeng': {'Lembaran': 40000, 'Jadi': 55000},
                'Selendang': {'Motif': 25000, 'Polos': 15000}
            },
            'Wanita': {
                'Kebaya': {'Brokat': 100000, 'Sari': 90000},
                'Kamen': {'Lembaran': 110000, 'Jadi': 150000},
                'Selendang': {'Motif': 25000, 'Polos': 15000}
            }
        }

        self.state = 'q0'
        self.gender = 'M'
        self.sizes = ['M', 'L', 'XL']
        self.total_harga = 0

    def hide_content(self):
        for widget in self.winfo_children():
            widget.place_forget()

    def show_welcome_screen(self):
        self.pilihan_barang = []

        image = Image.open("image\cover.png")
        resized_image = image.resize((500, 350),
Image.Resampling.LANCZOS)
```

```

self.bg_image = ImageTk.PhotoImage(resized_image)

image = tk.Label(self, image=self.bg_image, bg="white")
image.place(x=0, y=0)

# Tampilkan layar selamat datang
welcome_label = tk.Label(self, text="Selamat datang di Vending
Machine \nBaju Adat Otomatis", font=("Montserrat", 16), bg="#f5f5f5")
welcome_label.place(x=80, y=30)

start_button = tk.Button(self, text="Mulai", width=12,
command=self.size_selection)
start_button.place(x=200, y=300)

def size_selection(self):
    self.state = 'q1'
    # Sembunyikan elemen dari layar selamat datang
    self.hide_content()

    # Tampilkan elemen untuk memilih ukuran baju
    size_label = tk.Label(self, text="Silahkan pilih ukuran baju
anda", font=("Montserrat", 16), bg="white")
    size_label.place(x=110, y=130)
    # Buat button
    for i, size in enumerate(self.sizes):
        button = tk.Button(
            self,
            text=size,
            width=12,
            command=lambda s=size: self.gender_selection(s),
        )
        button.place(x=105 + i * 100, y=180)

def gender_selection(self, s):
    self.ukuran = s
    if self.ukuran == 'M':
        self.state = 'q2'
    elif self.ukuran == 'L':
        self.state = 'q3'
    elif self.ukuran == 'XL':
        self.state = 'q4'
    # Sembunyikan elemen dari layar size selection
    self.hide_content()

```

```

        # Tampilkan elemen untuk memilih gender
        gender_label = tk.Label(self, text="Untuk siapa baju ini?",
font=("Montserrat", 16), bg="white")
        gender_label.place(x=150, y=80)
        # Input gambar
        image_f = Image.open("image\wanita.png")
        image_m = Image.open("image\pria.png")
        # Resize gambar
        resized_image_f = image_f.resize((90, 90),
Image.Resampling.LANCZOS)
        self.tk_image_f = ImageTk.PhotoImage(resized_image_f)
        resized_image_m = image_m.resize((90, 90),
Image.Resampling.LANCZOS)
        self.tk_image_m = ImageTk.PhotoImage(resized_image_m)
        # Tampilkan gambar
        image_f_label = tk.Label(self, image=self.tk_image_f,
bg="white")
        image_f_label.place(x=145, y=120)
        image_m_label = tk.Label(self, image=self.tk_image_m,
bg="white")
        image_m_label.place(x=265, y=120)
        # Buat button
        gender_F_button = tk.Button(self, text="Wanita", width=12,
command=self.female_select)
        gender_F_button.place(x=145, y=225)
        gender_M_button = tk.Button(self, text="Pria", width=12,
command=self.male_select)
        gender_M_button.place(x=265, y=225)

    def female_select(self):
        self.state = 'q6'
        # Sembunyikan elemen dari layar gender selection
        self.hide_content()

        self.gender = 'F'
        #Tampilkan pilihan barang yang ingin dibeli
        female_label = tk.Label(self, text="Pakaian apa yang ingin anda
beli?", font=("Montserrat", 16), bg="white")
        female_label.place(x=90, y=80)

        # Akses gambar
        kebaya_img = Image.open("image\kebaya.png")

```

```

        kamen_img = Image.open("image\kamen_f.png")
        selendang_img = Image.open("image\selendang_f.png")
        # Atur ukuran gambar
        resized_kebaya = kebaya_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.kebaya_image = ImageTk.PhotoImage(resized_kebaya)
        resized_kamen = kamen_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.kamen_image = ImageTk.PhotoImage(resized_kamen)
        resized_selendang = selendang_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.selendang_image = ImageTk.PhotoImage(resized_selendang)
        # Tempatkan gambar pada GUI
        kebaya_img = tk.Label(self, image=self.kebaya_image,
bg="white")
        kebaya_img.place(x=90, y=120)
        kamen_img = tk.Label(self, image=self.kamen_image, bg="white")
        kamen_img.place(x=205, y=120)
        selendang_img = tk.Label(self, image=self.selendang_image,
bg="white")
        selendang_img.place(x=320, y=120)
        # Buat button
        kebaya_button = tk.Button(self, text="Kebaya", width=12,
command=self.kebaya_select)
        kebaya_button.place(x=90, y=235)
        kamen_button = tk.Button(self, text="Kamen", width=12,
command=self.kamen_select)
        kamen_button.place(x=205, y=235)
        selendang_button = tk.Button(self, text="Selendang", width=12,
command=self.selendang_f_select)
        selendang_button.place(x=320, y=235)

    def male_select(self):
        self.state = 'q5'
        # Sembunyikan elemen dari layar gender selection
        self.hide_content()

        self.gender = 'M'
        # Tampilkan pilihan barang yang ingin dibeli
        male_label = tk.Label(self, text="Pakaian apa yang ingin anda
beli?", font=("Montserrat", 16), bg="white")
        male_label.place(x=90, y=80)
        # Akses gambar

```



```

        baju_img = Image.open("image\safari.png")
        udeng_img = Image.open("image\m_udeng.png")
        kamen_img = Image.open("image\kamen_m.png")
        selendang_img = Image.open("image\selendang_m_polos.png")
        # Atur ukuran gambar
        resized_baju = baju_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.baju_image = ImageTk.PhotoImage(resized_baju)
        resized_udeng = udeng_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.udeng_image = ImageTk.PhotoImage(resized_udeng)
        resized_kamen = kamen_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.kamen_image = ImageTk.PhotoImage(resized_kamen)
        resized_selendang = selendang_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.selendang_image = ImageTk.PhotoImage(resized_selendang)
        # Tempatkan gambar pada GUI
        baju_img = tk.Label(self, image=self.baju_image, bg="white")
        baju_img.place(x=50, y=120)
        udeng_img = tk.Label(self, image=self.udeng_image, bg="white")
        udeng_img.place(x=150, y=120)
        kamen_img = tk.Label(self, image=self.kamen_image, bg="white")
        kamen_img.place(x=250, y=120)
        selendang_img = tk.Label(self, image=self.selendang_image,
bg="white")
        selendang_img.place(x=350, y=120)
        # Buat button
        baju_button = tk.Button(self, text="Baju Adat", width=12,
command=self.baju_adat_select)
        baju_button.place(x=50, y=235)
        udeng_button = tk.Button(self, text="Udeng", width=12,
command=self.udeng_select)
        udeng_button.place(x=150, y=235)
        kamen_button = tk.Button(self, text="Kamen & Saput", width=12,
command=self.kamen_saput_select)
        kamen_button.place(x=250, y=235)
        selendang_button = tk.Button(self, text="Selendang", width=12,
command=self.selendang_m_select)
        selendang_button.place(x=350, y=235)

    def kebaya_select(self):
        self.state = 'q11'

```

```

        # Sembunyikan elemen dari layar female select
        self.hide_content()

        # Tampilkan pilihan kebaya
        kebaya_label = tk.Label(self, text="Kebaya jenis apa yang ingin
anda beli?", font=("Montserrat", 14), bg="white")
        kebaya_label.place(x=90, y=75)

        # Input gambar
        brokat_img = Image.open("image\kebaya.png")
        sari_img = Image.open("image\sari.png")
        # Resize gambar
        resized_brokat = brokat_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_brokat = ImageTk.PhotoImage(resized_brokat)
        resized_sari = sari_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_sari = ImageTk.PhotoImage(resized_sari)
        # Tampilkan gambar
        brokat_label = tk.Label(self, image=self.image_brokat,
bg="white")
        brokat_label.place(x=140, y=120)
        sari_label = tk.Label(self, image=self.image_sari, bg="white")
        sari_label.place(x=260, y=120)
        # Buat button
        brokat_button = tk.Button(self, text="Kebaya Brokat", width=12,
command=lambda: self.item_select('Wanita', 'Kebaya', 'Brokat',
self.harga_barang['Wanita']['Kebaya']['Brokat']))
        brokat_button.place(x=140, y=230)
        sari_button = tk.Button(self, text="Kebaya Sari", width=12,
command=lambda: self.item_select('Wanita', 'Kebaya', 'Sari',
self.harga_barang['Wanita']['Kebaya']['Sari']))
        sari_button.place(x=260, y=230)
        # Isi harga
        brokat_price_label = tk.Label(self, text="Rp100.000,00",
font=("Montserrat", 8), bg="white")
        brokat_price_label.place(x=150, y=260)
        sari_price_label = tk.Label(self, text="Rp90.000,00",
font=("Montserrat", 8), bg="white")
        sari_price_label.place(x=270, y=260)

    def kamen_select(self):
        self.state = 'q12'

```

```

        # Sembunyikan elemen dari layar female select
        self.hide_content()

        # Tampilkan pilihan kamen
        kamen_label = tk.Label(self, text="Kamen jenis apa yang ingin
anda beli?", font=("Montserrat", 14), bg="white")
        kamen_label.place(x=90, y=75)
        # Input gambar
        kamen_img = Image.open("image\kamen_f.png")
        lembar_img = Image.open("image\kamen_lembar.png")
        # Resize gambar
        resized_kamen = kamen_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_kamen = ImageTk.PhotoImage(resized_kamen)
        resized_lembar = lembar_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_lembar = ImageTk.PhotoImage(resized_lembar)
        # Tampilkan gambar
        kamen_label = tk.Label(self, image=self.image_kamen,
bg="white")
        kamen_label.place(x=140, y=120)
        lembar_label = tk.Label(self, image=self.image_lembar,
bg="white")
        lembar_label.place(x=260, y=120)
        # Buat button
        kamen_jadi_button = tk.Button(self, text="Kamen Jadi",
width=12, command=lambda: self.item_select('Wanita', 'Kamen', 'Jadi',
self.harga_barang['Wanita']['Kamen']['Jadi']))
        kamen_jadi_button.place(x=140, y=230)
        kamen_lembar_button = tk.Button(self, text="Kamen Lembar",
width=12, command=lambda: self.item_select('Wanita', 'Kamen',
'Lembaran', self.harga_barang['Wanita']['Kamen']['Lembaran']))
        kamen_lembar_button.place(x=260, y=230)
        # Isi harga
        kamen_price_label = tk.Label(self, text="Rp150.000,00",
font=("Montserrat", 8), bg="white")
        kamen_price_label.place(x=150, y=260)
        lembar_price_label = tk.Label(self, text="Rp110.000,00",
font=("Montserrat", 8), bg="white")
        lembar_price_label.place(x=270, y=260)

    def selendang_f_select(self):
        self.state = 'q13'

```

```

        # Sembunyikan elemen dari layar female select
        self.hide_content()

        # Tampilkan pilihan selendang
        selendang_label = tk.Label(self, text="Selendang jenis apa yang
ingin anda beli?", font=("Montserrat", 14), bg="white")
        selendang_label.place(x=80, y=75)
        # Input gambar
        motif_img = Image.open("image\selendang_f.png")
        polos_img = Image.open("image\selendang_polos.png")
        # Resize gambar
        resized_motif = motif_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_motif = ImageTk.PhotoImage(resized_motif)
        resized_polos = polos_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_polos = ImageTk.PhotoImage(resized_polos)
        # Tampilkan gambar
        motif_label = tk.Label(self, image=self.image_motif,
bg="white")
        motif_label.place(x=140, y=120)
        polos_label = tk.Label(self, image=self.image_polos,
bg="white")
        polos_label.place(x=260, y=120)
        # Buat button
        selendang_motif_button = tk.Button(self, text="Motif",
width=12, command=lambda: self.item_select('Wanita', 'Selendang',
'Motif', self.harga_barang['Wanita']['Selendang']['Motif']))
        selendang_motif_button.place(x=140, y=230)
        selendang_polos_button = tk.Button(self, text="Polos",
width=12, command=lambda: self.item_select('Wanita', 'Selendang',
'Polos', self.harga_barang['Wanita']['Selendang']['Polos']))
        selendang_polos_button.place(x=260, y=230)
        # Input harga
        motif_price_label = tk.Label(self, text="Rp25.000,00",
font=("Montserrat", 8), bg="white")
        motif_price_label.place(x=150, y=260)
        polos_price_label = tk.Label(self, text="Rp15.000,00",
font=("Montserrat", 8), bg="white")
        polos_price_label.place(x=270, y=260)

    def baju_adat_select(self):
        self.state = 'q7'

```

```

        # Sembunyikan elemen dari layar male select
        self.hide_content()

        # Tampilkan pilihan selendang
        baju_adat_label = tk.Label(self, text="Baju adat jenis apa yang
ingin anda beli?", font=("Montserrat", 14), bg="white")
        baju_adat_label.place(x=80, y=75)
        # Input gambar
        safari_img = Image.open("image\safari.png")
        kemeja_img = Image.open("image\kemeja.png")
        # Resize gambar
        resized_safari = safari_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_safari = ImageTk.PhotoImage(resized_safari)
        resized_kemeja = kemeja_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_kemeja = ImageTk.PhotoImage(resized_kemeja)
        # Tampilkan gambar
        safari_label = tk.Label(self, image=self.image_safari,
bg="white")
        safari_label.place(x=140, y=120)
        kemeja_label = tk.Label(self, image=self.image_kemeja,
bg="white")
        kemeja_label.place(x=260, y=120)
        # Buat button
        safari_button = tk.Button(self, text="Safari", width=12,
command=lambda: self.item_select('Pria', 'Baju Adat', 'Safari',
self.harga_barang['Pria']['Baju Adat']['Safari']))
        safari_button.place(x=140, y=230)
        kemeja_button = tk.Button(self, text="Kemeja", width=12,
command=lambda: self.item_select('Pria', 'Baju Adat', 'Kemeja',
self.harga_barang['Pria']['Baju Adat']['Kemeja']))
        kemeja_button.place(x=260, y=230)
        # Input harga
        safari_price_label = tk.Label(self, text="Rp100.000,00",
font=("Montserrat", 8), bg="white")
        safari_price_label.place(x=150, y=260)
        kemeja_price_label = tk.Label(self, text="Rp90.000,00",
font=("Montserrat", 8), bg="white")
        kemeja_price_label.place(x=270, y=260)

    def udeng_select(self):
        self.state = 'q8'

```

```

        # Sembunyikan elemen dari layar male select
        self.hide_content()

        # Tampilkan pilihan udeng
        udeng_label = tk.Label(self, text="Udeng jenis apa yang ingin
anda beli?", font=("Montserrat", 14), bg="white")
        udeng_label.place(x=80, y=75)
        # Input gambar
        udeng_jadi_img = Image.open("image\m_udeng_jadi.png")
        udeng_lembar_img = Image.open("image\m_udeng_lembar.png")
        # Resize gambar
        resized_udeng_jadi = udeng_jadi_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_udeng_jadi = ImageTk.PhotoImage(resized_udeng_jadi)
        resized_udeng_lembar = udeng_lembar_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_udeng_lembar =
ImageTk.PhotoImage(resized_udeng_lembar)
        # Tampilkan gambar
        udeng_jadi_label = tk.Label(self, image=self.image_udeng_jadi,
bg="white")
        udeng_jadi_label.place(x=140, y=120)
        udeng_lembar_label = tk.Label(self,
image=self.image_udeng_lembar, bg="white")
        udeng_lembar_label.place(x=260, y=120)
        # Buat button
        udeng_jadi_button = tk.Button(self, text="Udeng Jadi",
width=12, command=lambda: self.item_select('Pria', 'Udeng', 'Jadi',
self.harga_barang['Pria']['Udeng']['Jadi']))
        udeng_jadi_button.place(x=140, y=230)
        udeng_lembar_button = tk.Button(self, text="Udeng Lembaran",
width=12, command=lambda: self.item_select('Pria', 'Udeng', 'Lembaran',
self.harga_barang['Pria']['Udeng']['Lembaran']))
        udeng_lembar_button.place(x=260, y=230)
        # Input harga
        jadi_price_label = tk.Label(self, text="Rp55.000,00",
font=("Montserrat", 8), bg="white")
        jadi_price_label.place(x=150, y=260)
        lembar_price_label = tk.Label(self, text="Rp40.000,00",
font=("Montserrat", 8), bg="white")
        lembar_price_label.place(x=270, y=260)

    def kamen_saput_select(self):

```

```

        self.state = 'q9'
        # Sembunyikan elemen dari layar male select
        self.hide_content()

        # Tampilkan pilihan selendang
        kamen_label = tk.Label(self, text="Kamen & saput jenis apa yang
ingin anda beli?", font=("Montserrat", 14), bg="white")
        kamen_label.place(x=60, y=75)
        # Input gambar
        kamen_jadi_img = Image.open("image\\kamen_m.png")
        kamen_lembar_img = Image.open("image\\kamen_lembar_m.png")
        # Resize gambar
        resized_kamen_jadi = kamen_jadi_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_kamen_jadi = ImageTk.PhotoImage(resized_kamen_jadi)
        resized_kamen_lembar = kamen_lembar_img.resize((90, 90),
Image.Resampling.LANCZOS)
        self.image_kamen_lembar =
ImageTk.PhotoImage(resized_kamen_lembar)
        # Tampilkan gambar
        kamen_jadi_label = tk.Label(self, image=self.image_kamen_jadi,
bg="white")
        kamen_jadi_label.place(x=140, y=120)
        kamen_lembar_label = tk.Label(self,
image=self.image_kamen_lembar, bg="white")
        kamen_lembar_label.place(x=260, y=120)
        # Buat button
        kamen_jadi_button = tk.Button(self, text="Setengah Jadi",
width=12, command=lambda: self.item_select('Pria', 'Kamen', 'Setengah
Jadi', self.harga_barang['Pria']['Kamen']['Setengah Jadi']))
        kamen_jadi_button.place(x=140, y=230)
        kamen_setjadi_button = tk.Button(self, text="Lembaran",
width=12, command=lambda: self.item_select('Pria', 'Kamen', 'Lembaran',
self.harga_barang['Pria']['Kamen']['Lembaran']))
        kamen_setjadi_button.place(x=260, y=230)
        # Input harga
        jadi_price_label = tk.Label(self, text="Rp85.000,00",
font=("Montserrat", 8), bg="white")
        jadi_price_label.place(x=150, y=260)
        lembar_price_label = tk.Label(self, text="Rp80.000,00",
font=("Montserrat", 8), bg="white")
        lembar_price_label.place(x=270, y=260)

```

```

def selendang_m_select(self):
    self.state = 'q10'
    # Sembunyikan elemen dari layar male select
    self.hide_content()

    # Tampilkan pilihan selendang
    selendang_label = tk.Label(self, text="Selendang jenis apa yang
ingin anda beli?", font=("Montserrat", 14), bg="white")
    selendang_label.place(x=70, y=75)
    # Input gambar
    motif_img = Image.open("image\selendang_m_motif.png")
    polos_img = Image.open("image\selendang_m_polos.png")
    # Resize gambar
    resized_motif = motif_img.resize((90, 90),
Image.Resampling.LANCZOS)
    self.image_motif = ImageTk.PhotoImage(resized_motif)
    resized_polos = polos_img.resize((90, 90),
Image.Resampling.LANCZOS)
    self.image_polos = ImageTk.PhotoImage(resized_polos)
    # Tampilkan gambar
    motif_label = tk.Label(self, image=self.image_motif,
bg="white")
    motif_label.place(x=140, y=120)
    polos_label = tk.Label(self, image=self.image_polos,
bg="white")
    polos_label.place(x=260, y=120)
    # Buat button
    selendang_motif_button = tk.Button(self, text="Motif",
width=12, command=lambda: self.item_select('Pria', 'Selendang',
'Motif', self.harga_barang['Pria']['Selendang']['Motif']))
    selendang_motif_button.place(x=140, y=230)
    selendang_polos_button = tk.Button(self, text="Polos",
width=12, command=lambda: self.item_select('Pria', 'Selendang',
'Polos', self.harga_barang['Pria']['Selendang']['Polos']))
    selendang_polos_button.place(x=260, y=230)
    # Input harga
    motif_price_label = tk.Label(self, text="Rp25.000,00",
font=("Montserrat", 8), bg="white")
    motif_price_label.place(x=150, y=260)
    polos_price_label = tk.Label(self, text="Rp15.000,00",
font=("Montserrat", 8), bg="white")
    polos_price_label.place(x=270, y=260)

```



```

def item_select(self, jenis_kelamin, kategori, variasi, harga):
    self.pilihan_barang.append({'jenis_kelamin': jenis_kelamin,
    'kategori': kategori, 'variasi': variasi, 'ukuran': self.ukuran,
    'harga': harga})
    self.confirmation()

def confirmation(self):
    self.state = 'q14, q15'
    # Sembunyikan elemen dari layar male select
    self.hide_content()

    # Tampilkan pilihan konfirmasi
    konfirmasi_label = tk.Label(self, text="Apakah anda sudah
selesai belanja?", font=("Montserrat", 14), bg="white")
    konfirmasi_label.place(x=90, y=140)

    ya_button = tk.Button(self, text="Ya", width=12,
command=self.nota)
    ya_button.place(x=140, y=180)

    tidak_button = tk.Button(self, text="Tidak", width=12,
command=self.choice)
    tidak_button.place(x=260, y=180)

def choice(self):
    if self.gender == "M":
        self.male_select()
    else:
        self.female_select()

def nota(self):
    self.state = 'q16'
    # Sembunyikan elemen dari layarsebelumnya
    self.hide_content()

    # Buat frame
    self.nota_frame = tk.Frame(self, bg="white")
    self.nota_frame.pack(side='left', padx=30, pady=20)
    self.qr_frame = tk.Frame(self, bg="white")
    self.qr_frame.pack(side='right', padx=30, pady=20)

    # Menampilkan barang-barang yang dibeli

```

```

        nota_label = tk.Label(self.nota_frame, text="Nota Pembelian",
font=("Montserrat", 14), bg="white")
        nota_label.pack(pady=7)

        for pembelian in self.pilihan_barang:
            detail_barang = f"{pembelian['jenis_kelamin']} -
{pembelian['kategori']} ({pembelian['variasi']}, Ukuran:
{pembelian['ukuran']}) : Rp{pembelian['harga']}"
            detail_barang_label = tk.Label(self.nota_frame,
text=detail_barang, font=("Montserrat", 8), bg="white")
            detail_barang_label.pack(anchor='w', pady=4)

        # Menampilkan total harga
        self.total_harga = sum(pembelian['harga'] for pembelian in
self.pilihan_barang)
        total_harga_label = tk.Label(self.nota_frame, text=f"Total
Harga: Rp{self.total_harga}", font=("Montserrat", 10), bg="white")
        total_harga_label.pack(anchor='w', pady=7)

        qr_img = Image.open("image\qr.png")
        # Resize gambar
        resized_qr = qr_img.resize((140, 140),
Image.Resampling.LANCZOS)
        self.qr_image = ImageTk.PhotoImage(resized_qr)
        qr_label = tk.Label(self.qr_frame, image=self.qr_image)
        qr_label.pack(pady=10)

        # Tombol pembayaran
        bayar_button = tk.Button(self.qr_frame, text="Bayar", width=12,
command=self.finish)
        bayar_button.pack(pady=10)

    def finish(self):
        self.state = 'q17'
        # Sembunyikan elemen dari layarsebelumnya
        for widget in self.wininfo_children():
            widget.pack_forget()

        image = Image.open("image\cover.png")
        resized_image = image.resize((500, 350),
Image.Resampling.LANCZOS)
        self.bg_image = ImageTk.PhotoImage(resized_image)

```

```

        image = tk.Label(self, image=self.bg_image, bg="white")
        image.place(x=0, y=0)

        goodbye_label = tk.Label(self, text="Terima kasih telah
berbelanja di\nVending Machine Baju Adat Otomatis", font=("Montserrat",
14), bg="#f5f5f5")
        goodbye_label.place(x=80, y=30)

        ambil_button = tk.Button(self, text="Ambil Barang", width=12,
command=self.show_welcome_screen)
        ambil_button.place(x=200, y=300)

    def run(self):
        self.mainloop()

if __name__ == "__main__":
    vending_machine_gui = GUI()
    vending_machine_gui.run()

```

3.4 Pengujian *Vending Machine*

Pengujian *vending machine* adalah langkah penting dalam pengembangan mesin penjual otomatis untuk memastikan bahwa sistem berfungsi dengan baik dan memenuhi kebutuhan *user*. Berikut adalah beberapa langkah yang dapat diambil dalam pengujian *vending machine*:

1. Pengujian Fungsional

Pengujian ini dilakukan dengan memastikan bahwa setiap alur sistem berfungsi dengan benar dan dapat mendistribusikan produk sesuai pesanan. Uji sistem pembayaran, seperti menampilkan *QR Code* dan nota yang dihasilkan oleh program yang telah merespon dengan sesuai. Pemeriksaan terhadap program dan sistem yang dibangun seperti pada program *vending machine* penjualan busana adat otomatis dengan melakukan metode wawancara kepada 5 orang *user*, dan melakukan validasi bahwa program dapat berfungsi dengan baik.

2. Pengujian Antarmuka *User*

Pengujian ini dilakukan dengan melakukan verifikasi terhadap tampilan layar dan tombol-tombol untuk memastikan antarmuka *user* mudah dipahami dan berfungsi dengan baik. pengujian interaksi dengan layar sentuh pengujian ini dilakukan ketika 5

orang *user* mencoba GUI tersebut dan program berjalan dengan baik sesuai dengan antarmuka yang ditampilkan.

BAB IV

MANUAL APLIKASI

4.1 Fitur Sistem

Sistem *vending machine* baju adat otomatis ini memiliki beberapa fitur, yakni:

1. Fitur pemilihan jenis pakaian

Sistem memberikan beberapa variasi produk, khususnya pakaian, sesuai dengan beberapa pilihan *user*, mulai dari jenis pakaian (wanita atau pria), ukuran, kategori barang (untuk wanita ada kebaya, kamen, dan selendang, sementara untuk pria ada baju adat, udeng, kamen dan saput, serta selendang), serta variasi barang sesuai dari masing-masing kategori.

2. Fitur penampilan nota

Sistem akan menampilkan nota untuk *user* yang berisi daftar barang-barang yang dibeli oleh *user* serta total harga yang harus dibayar *user*.

3. Fitur pembayaran

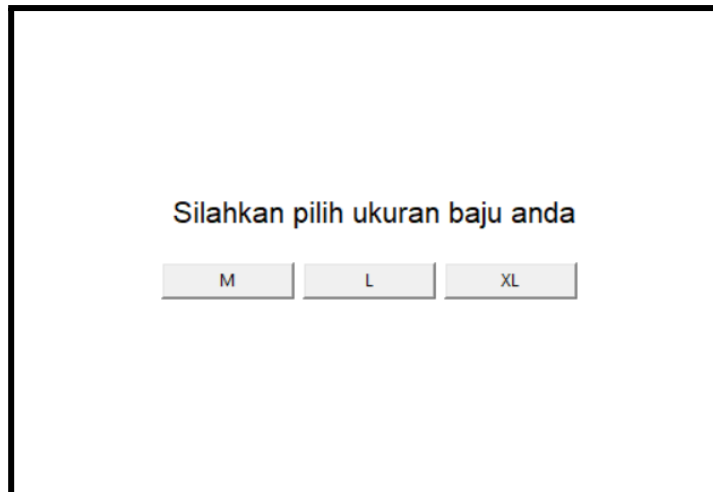
Sistem menyediakan fitur pembayaran *cashless* berupa *scan* Qris. Pada penampilan nota, akan ditampilkan pula Qris sebagai metode pembayaran.

4.2 Antarmuka



Gambar 9.Tampilan Awal Desain GUI

Ini merupakan tampilan pertama yang dilihat oleh *user* pada layar utama *vending machine* kami. Pada bagian atas layar menampilkan pesan selamat datang kepada *user* dan suatu *button* dengan *text* ‘Mulai’ yang dapat diklik oleh *user* untuk memulai proses belanja pada *vending machine*.



Gambar 10. Pemilihan Ukuran Busana

Ketika *user* telah menekan *button* ‘Mulai’, maka tampil layar selanjutnya yang menginformasikan *user* untuk memilih ukuran pakaian adat yang ingin mereka beli. Ada pilihan ukuran M, L, dan XL yang diwakili oleh tiga *button* berbeda. *User* kemudian dapat menekan salah satu *button* yang ada.



Gambar 11. Pemilihan Gender

Setelah *user* memilih ukuran pakaian, selanjutnya layar menunjukkan pilihan *gender* atau jenis kelamin penerima baju. Ada pilihan wanita dan pria yang diwakili oleh *button* masing-masing. *User* kemudian dapat menekan salah satu *button* yang ada. Pilihan tersebut akan mengarahkan *user* ke layar yang sesuai dengan *gender* yang mereka pilih.

WANITA	PRIA
--------	------



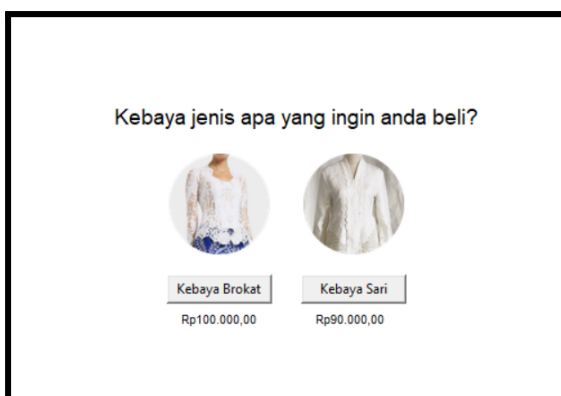
Gambar 12. Busana Adat Wanita

Apabila *user* menekan *button* 'Wanita', maka layar akan menampilkan pilihan kategori pakaian wanita yang ada pada *vending machine*. Ada tiga kategori pakaian wanita, yakni kebaya, kamen, dan selendang. Pada layar sendiri, telah terdapat pesan yang menanyakan kategori pakaian apa yang ingin dibeli oleh *user*. Ada pula tiga gambar dan *button* yang mewakili masing-masing pilihan.



Gambar 13. Busana Adat Pria

Apabila *user* menekan *button* 'Pria', maka layar akan menampilkan pilihan kategori pakaian pria yang ada pada *vending machine*. Ada tiga kategori pakaian wanita, yakni baju adat, udeng, kamen & saput, dan selendang. Pada layar sendiri, telah terdapat pesan yang menanyakan kategori pakaian apa yang ingin dibeli oleh *user*. Ada pula empat gambar dan *button* yang mewakili masing-masing pilihan.



Gambar 14. Jenis Kebaya

Apabila *user* memilih kategori kebaya, maka layar akan menampilkan variasi kebaya yang ada pada *vending machine*. Ada dua kategori kebaya, yakni kebaya



Gambar 15. Jenis Kemeja

Apabila *user* memilih kategori baju adat, maka layar akan menampilkan variasi baju adat yang ada pada *vending machine*. Ada dua kategori baju adat, yakni safari dan

brokat dan kebaya sari dengan gambar serta harga masing-masing. Ada pula dua *button* yang mewakili tiap variasi yang dapat ditekan *user* untuk memilih variasi yang *user* inginkan. Dengan memilih salah satu variasi, maka *user* akan diarahkan ke layar konfirmasi.

kemeja dengan gambar serta harga masing-masing. Ada pula dua *button* yang mewakili tiap variasi yang dapat ditekan *user* untuk memilih variasi yang *user* inginkan. Dengan memilih salah satu variasi, maka *user* akan diarahkan ke layar konfirmasi.



Gambar 16. Jenis Kamen

Apabila *user* memilih kategori kamen, maka layar akan menampilkan variasi kamen yang ada pada *vending machine*. Ada dua kategori kamen, yakni kamen jadi dan kamen lembaran dengan gambar serta harga masing-masing. Ada pula dua *button* yang mewakili tiap variasi yang dapat ditekan *user* untuk memilih variasi yang *user* inginkan. Dengan memilih salah satu variasi, maka *user* akan diarahkan ke layar konfirmasi.



Gambar 17. Jenis Udeng

Apabila *user* memilih kategori udeng, maka layar akan menampilkan variasi udeng yang ada pada *vending machine*. Ada dua kategori udeng, yakni udeng jadi dan udeng lembaran dengan gambar serta harga masing-masing. Ada pula dua *button* yang mewakili tiap variasi yang dapat ditekan *user* untuk memilih variasi yang *user* inginkan. Dengan memilih salah satu variasi, maka *user* akan diarahkan ke layar konfirmasi.



Gambar 18. Jenis Selendang Wanita

Apabila *user* memilih kategori selendang, maka layar akan menampilkan variasi selendang yang ada pada *vending machine*. Ada dua kategori selendang, yakni selendang motif dan selendang polos dengan gambar serta harga masing-masing. Ada pula dua *button* yang mewakili tiap variasi yang dapat ditekan *user* untuk memilih variasi yang *user* inginkan. Dengan memilih salah satu variasi, maka *user* akan diarahkan ke layar konfirmasi.



Gambar 19. Jenis Kamen dan Saput

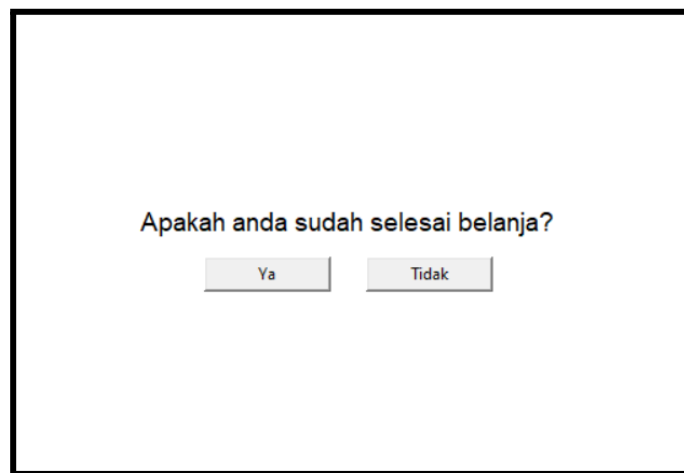
Apabila *user* memilih kategori kamen dan saput, maka layar akan menampilkan variasi kamen dan saput yang ada pada *vending machine*. Ada dua kategori kamen dan saput, yakni kamen dan saput setengah jadi serta kamen dan saput lembaran dengan gambar serta harga masing-masing. Ada pula dua *button* yang mewakili tiap variasi yang dapat ditekan *user* untuk memilih variasi yang *user* inginkan. Dengan memilih salah satu variasi, maka *user* akan diarahkan ke layar konfirmasi.



Gambar 20. Jenis Selendang Pria

Apabila *user* memilih kategori selendang, maka layar akan menampilkan variasi selendang yang ada pada *vending machine*.

	Ada dua kategori selendang, yakni selendang motif dan selendang polos dengan gambar serta harga masing-masing. Ada pula dua <i>button</i> yang mewakili tiap variasi yang dapat ditekan <i>user</i> untuk memilih variasi yang <i>user</i> inginkan. Dengan memilih salah satu variasi, maka <i>user</i> akan diarahkan ke layar konfirmasi.
--	--



Gambar 21. Proses Belanja

Ini merupakan layar konfirmasi. Setelah *user* memilih variasi kategori pakaian yang mereka inginkan, maka layar akan menunjukkan pesan yang menanyakan apakah *user* sudah selesai belanja. Apabila *user* menekan *button* 'Tidak', maka layar akan menunjukkan layar pilihan kategori untuk *gender* yang mereka pilih. Apabila *user* menekan *button* 'Ya', maka layar akan beralih ke layar nota pembelian dan pembayaran.



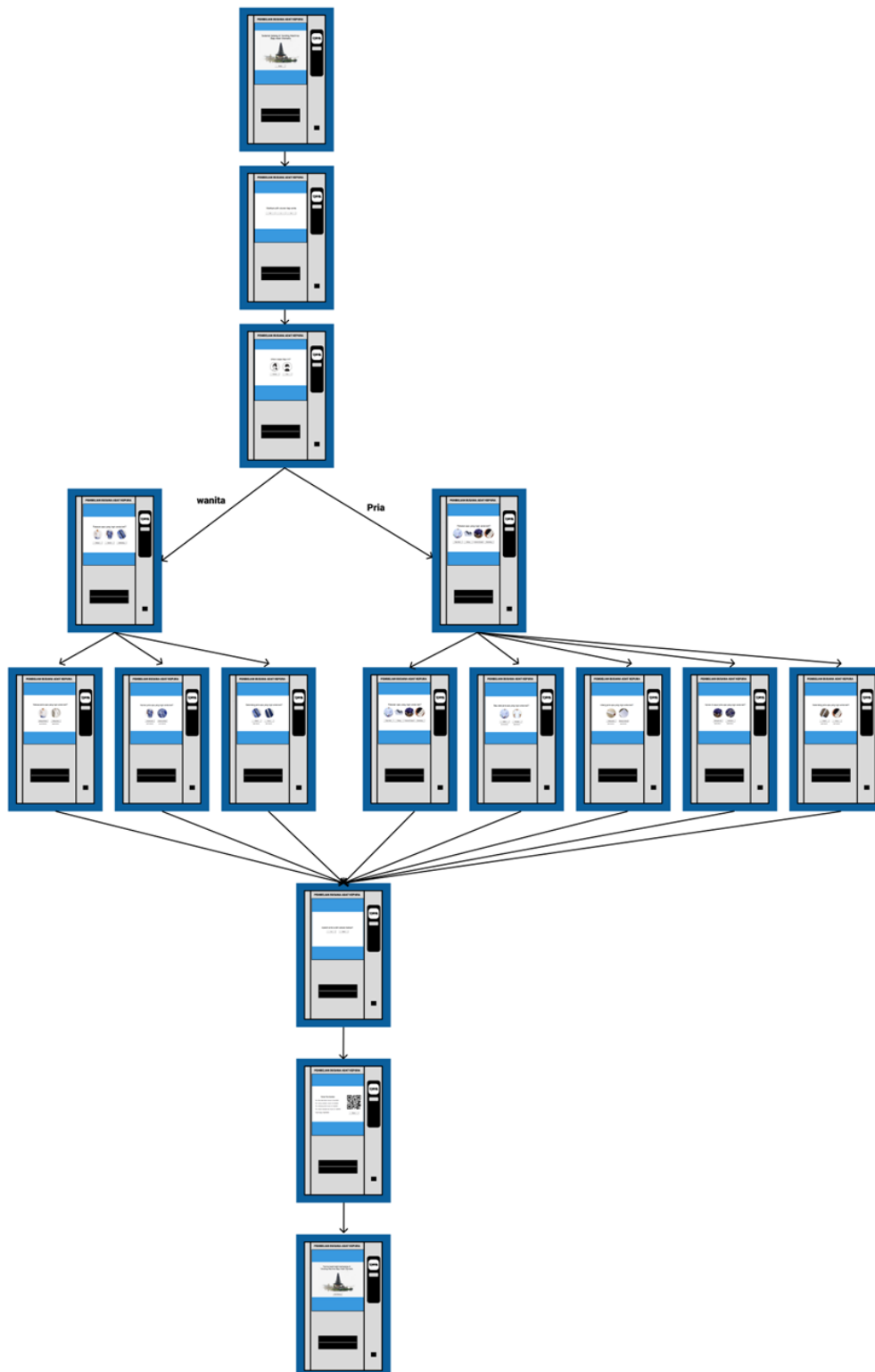
Gambar 22. Nota Pembelian dan Pembayaran

Ini merupakan layar nota pembelian dan pembayaran. Apabila *user* telah memberikan konfirmasi bahwa *user* selesai belanja, maka nota pembelian yang terdiri dari daftar barang-barang yang *user* beli serta total harganya akan ditampilkan. Selain itu, ditampilkan pula Qris yang dapat *user scan* untuk melakukan pembayaran. Pada GUI ini, proses *scanning* tersebut kami wakikan dengan *button* 'Bayar'. Ketika *user* menekan *button* tersebut, maka *vending machine* akan menampilkan layar selanjutnya.



Gambar 23. Ambil Barang

Layar ini merupakan layar terakhir yang menampilkan pesan terima kasih kepada *user* karena telah berbelanja di *vending machine* baju adat otomatis dan *button* 'Ambil Barang' yang mewakilkan proses *user* mengambil barang yang mereka beli dari *vending machine*. Ketika *user* menekan *button* tersebut, maka *vending machine* akan menampilkan layar awal sebelum proses belanja dimulai.



Gambar 24. Desain Sistem

Di atas ini merupakan desain *vending machine* secara keseluruhan dengan badan mesinnya sendiri. Telah ditampilkan pula alur sistem dari *vending machine* baju adat otomatis ini.

BAB V

PENUTUP

5.1 Kesimpulan

Dalam laporan ini, kami merencanakan *Vending Machine* untuk otomatisasi penjualan busana adat di objek wisata Pura Tampak Siring, Bali. Kami mengidentifikasi masalah utama, yaitu kurangnya persiapan pengunjung dengan pakaian yang pantas serta ketergantungan pada tenaga manusia dalam penjualan busana adat. Untuk mengatasi tantangan ini, kami mengusulkan solusi berupa mesin penjualan otomatis yang menggunakan konsep *Finite State Automata* (FSA).

Kami merumuskan permasalahan, tujuan, asumsi, dan batasan yang terkait dengan perancangan *Vending Machine* ini. Kami juga menguraikan proses perancangan FSA untuk mengatur alur pembelian dan pembayaran, serta merinci diagram aktivitas yang menjelaskan langkah-langkah *user* dalam menjalankan aplikasi mesin. Selain itu, kami membahas implementasi algoritma FSA dengan menggunakan bahasa pemrograman Python.

kesimpulannya, kami berhasil merancang solusi yang mengatasi permasalahan persiapan busana adat bagi pengunjung dan mengurangi ketergantungan pada waktu operasional manusia. Dengan adanya *Vending Machine* ini, pengunjung dapat dengan mudah membeli busana adat secara otomatis, meningkatkan kenyamanan mereka dan tetap menghormati adat setempat di Bali.

LAMPIRAN

Folder program dan tugas lengkap :

https://drive.google.com/drive/folders/1njirjoYfDnzDlrfa66ZqOJWvs1pCz_mq?usp=sharing

Link diagram transisi :

<https://whimsical.com/matdis-86TYJNNQVeAohmXYc7EK7u>

Link tabel transisi :

<https://docs.google.com/spreadsheets/d/1-PQsLvonseS8Jb89Rpfo8-EqSqDw7HW-jCNQlv1D-mQ/edit?usp=sharing>

Link video :

<https://youtu.be/1m0JbzVw1Mw>