BAB 3

ANALISIS DAN PERANCANGAN

Pada bab ini menjelaskan mengenai analisis dan gambaran dari sistem yang ada dan bagaimana perancangan program aplikasi yang akan dibuat.

3.1 Analisis Masalah

Analisis masalah merupakan proses identifikasi serta evaluasi mengenai masalah-masalah yang timbul dari pengembangan sistem, baik mengenai aturan serta teknik-teknik yang digunakan dalam pengembangan sistem.

Ada beberapa permainan yang menjadi rujukan dalam membuat permainan *shopping time* ini, yaitu:

1. *Brain Spa Visual Memori* 2

Permainan ini bisa dilihat di situs Brain Spa Visual Memori 2 [16]. Dalam permainan ini, pemain diajak untuk berbelanja produk-produk sehari-hari, mulai dari berbagai macam produk *baby care, snack, canned goods* dan lain-lain. Aturan permainan ini adalah sebagai berikut:

- Misi permainnan ini adalah pemain harus membeli produk sesuai dengan uang yang diberikan.
- Satu macam produk hanya boleh dibeli satu saja.
- Jika misi terpenuhi maka akan lanjut ke level berikutnya.

Keunggulan permainan ini adalah adanya nilai edukasi yang terlihat dari pemain harus memilih barang yang mana yang harus dia beli agar uang yang ada habis tanpa sisa. Namun permainan ini juga memiliki kelemahan yaitu karakter yang ada dalam permainan ini hanya bisa berdiri di tempat yang sama, selain itu dalam permainan ini tidak ada NPC (*Non Playable Character*) yang menjadi musuh bagi pemain dalam menyelesaikan misi yang diberikan sehingga permainan menjadi sedikit membosankan.

Berikut adalah tampilan permainan *Brain Spa Visual Memori* 2.



Gambar 3.1 Tampilan Menu Awal Permainan *Brain Spa Visual Memori* 2 [16]



Gambar 3.2 Tampilan Permainan *Brain Spa Visual Memori* 2 [16]

2. *Shopping Street*

Permainan ini dapat dilihat di situs Shopping Street [17]. Aturan yang berlaku dalam permainan adalah:

- Misi permainan ini adalah pemain bisa menghasilkan uang sesuai dengan yang telah ditetapkan tiap levelnya dengan membangun toko-toko dan hal-hal pendukung lainnya
- Diawal permainan, pemain diberikan modal uang untuk bisa membangun satu toko.
- Uang yang diperoleh berasal dari kunjungan NPC ke toko dan membeli barang yang ada di toko tersebut.
- Satu level dibagi menjadi 5 hari kerja.
- Pemain bisa menambah toko kapan saja jika uang yang ada mencukupi.

Kekurangan dari permainan ini adalah tidak adanya karakter yang bisa dimainkan oleh pemain untuk melakukan transaksi belanja. Berikut adalah tampilan dari permainan *shopping street*:



Gambar 3.3 Tampilan Menu Utama Permainan *Shopping Street* [17]



Gambar 3.4 Tampilan Permainan *Shopping Street* [17]

3. *Shopping Corner*

Permainan ini bisa dimainkan di situs Shopping Corner [18]. Aturan permainan adalah sebagai berikut:

- Misi permainan ini adalah pemain bisa menarik perhatian pengunjung untuk mengunjungi toko yang dibangun, tiap level harus ada 100 pengunjung yang mengunjungi toko.
- Diawal permainan, pemain diberikan modal uang untuk bisa membangun satu toko.
- Uang yang diperoleh berasal dari kunjungan NPC ke toko dan membeli barang yang ada di toko tersebut.
- Satu level dibagi menjadi 5 hari kerja.
- Pemain bisa menambah toko kapan saja jika uang yang ada mencukupi dan tiap toko hanya bisa dibangun satu saja.

Kelemahan permainan ini adalah kegiatan jual belinya tidak terlalu terlihat karena pemain tidak bisa mengatur karakter untuk berbelanja, selain itu dalam

permainan ini juga tidak ada nilai edukasi. Berikut adalah tampilan permainan *shopping corner*.



Gambar 3.5 Tampilan Menu Utama Permainan *Shopping Corner* [18]



Gambar 3.6 Tampilan Permainan *Shopping Corner* [18]

Dengan melihat adanya kekurangan dari permainan-permainan diatas, maka dibuatlah permainan *shopping time* yang akan menggabungkan kelebihan-kelebihan dan memperbaiki kekurangan-kekurangan dari permainan tersebut.

3.1.1 Pengenalan Aplikasi Permainan Shopping Time

Permainan *shopping time* merupakan permainan 2D (2 dimensi) yang mengambil tema tentang kegiatan berbelanja. Diawal permainan akan muncul

gambar pusat perbelanjaan, terdapat 6 toko yang pada masing-masing toko menjual 4 macam barang yang berbeda.

Misi permainan ini adalah menghabiskan uang jajan yang diberikan dengan membeli barang-barang yang ada dalam jangka waktu yang telah ditentukan. Terdapat karakter seorang anak perempuan yang merupakan wujud pengguna dalam aplikasi permainan *shopping time* dalam bentuk 2D.

Permainan *shopping time* ini merupakan permainan berbasis *desktop* yang bersifat *stand alone*. *Software* yang digunakan untuk membuat permainan ini adalah Adobe Flash (Action Script 2). Dalam permainan ini juga diterapkan algoritma A* yang pengunaannya hanya untuk menentukan jalan optimal yang diiimplementasikan pada karakter yang ada di permainan ini.

Aplikasi permainan ini ditujukan untuk anak usia 6-12 tahun karena adanya nilai edukasi yang diterapkan yaitu perhitungan sederhana dan pengenalan warna. Dalam permainan ini terdapat 3 level yang memiliki tingkat kesulitan yang berbeda, perbedaan tiap level terletak pada nilai uang jajan, waktu yang diberikan dan pertanyaan yang diberikan. Hal yang menarik dari aplikasi permainan *shopping time* ini selain adanya nilai edukasi yaitu perhitungan sederhana dan pengenalan warna, pada level 3 juga terdapat NPC yaitu seorang pencuri yang bisa mencuri barang yang ada di toko.

3.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional merupakan proses indentifikasi dan evaluasi mengenai kebutuhan sistem maupun pengguna, sehingga diharapkan permainan yang dibangun nanti dapat berjalan dengan baik.

Pada analisis kebutuhan sistem non fungsional ini dijelaskan mengenai analisis masukan, analisis keluaran, kebutuhan perangkat keras (*hardware),* kebutuhan perangkat lunak (*software),* dan pengguna (*user*) yang terdiri dari:

a. Analisis masukan

Analisis masukan dilakukan untuk mengetahui masukan data apa saja yang diperlukan oleh sistem. Masukan data yang diperlukan oleh sistem yaitu titik awal, titik tujuan, baris kotak, kolom kotak, penghalang maupun tanpa penghalang dan nilai heuristik.

b. Analisis keluaran

Analisis keluaran dilakukan untuk mengetahui keluaran dari sistem yang dibutuhkan oleh *user* yaitu berupa jalur terpendek dan pelacakan pencarian dari titik awal ke titik tujuan.

c. Analisis perangkat keras *(hardware)*

Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Kebutuhan perangkat keras ini bisa dibagi menjadi dua sudut pandang yaitu dari sisi pembangun dan pengguna. Tabel 3.2 adalah spesifikasi minimun perangkat keras yang dibutuhkan untuk membangun aplikasi permainan *shopping time* dan tabel 3.3 adalah spesifikasi minimun

perangkat keras yang dibutuhkan untuk menjalankan aplikasi permainan *shopping time* secara optimal.

Tabel 3.1 Spesifikasi Minimum Perangkat Keras Di Sisi Pembangun

Komputer	**Spesifikasi perangkat keras**
Prosesor	Prosesor 1.73 Ghz
Resolusi layar	1024 x 800 pixel
Memori	Memori 1 GB
Harddisk	Harddisk 120 GB
Monitor/LCD	-
Keyboard dan mouse	-

Tabel 3.2 Spesifikasi Minimum Perangkat Keras Di Sisi Pengguna

Komputer	**Spesifikasi perangkat keras**
Prosesor	Prosesor 1.6 Ghz
Resolusi layar	800 x 600 pixel
Memori	Memori 512 MB
Harddisk	Harddisk 1 GB
Monitor/LCD	-
Keyboard dan mouse	-

d. Analisis kebutuhan perangkat lunak (*software*)

Permainan *shopping time* memiliki kebutuhan perangkat lunak (*software*) pada tabel 3.9.

Tabel 3.3 Spesifikasi Perangkat Lunak Dari Sisi Pembuat

Komputer	**Spesifikasi perangkat lunak**
Sistem Operasi	Window
Bahasa Pemograman	Adobe Flash CS 4

Sedangkan dari sisi pengguna selain memiliki sistem operasi windows, PC (Personal Computer)/Laptop juga harus telah terinstal aplikasi flash player.

e. Analisis pengguna

Suatu aplikasi dapat berjalan optimal apabila pengguna memiliki kemampuan untuk menjalankan aplikasi yang bersangkutan. Untuk memainkan *game* setidaknya pengguna harus memiliki:

1. Usia : 6-12 tahun
2. Pendidikan : sekolah dasar
3. Keterampilan : mampu menggunakan petunjuk yang ada di aplikasi ini dan dapat menjalankan aplikasi pada PC (*Personal Computer*)
4. Pengalaman : pernah menggunakan komputer dan bermain *game* komputer.
5. Kemampuan berbahasa : bisa berbahasa indonesia.

3.2.1 Analisis Antar Muka

Pada permainan shopping time, di halaman utama terdapat 5 tombol, yaitu tombol mulai, tombol petunjuk, tombol pengaturan, tombol kredit dan tombol keluar.

Berikut merupakan gambar tampilan halaman utama dari aplikasi permainan *shopping time*:

Shopping Time	
Halaman Utama	
Visualisasi	**Keterangan**
	• Background merupakan toko-toko. Dihalaman ini terdapat 5 tombol yaitu: • Tombol mulai untuk masuk ke halaman game • Tombol petunjuk untuk masuk ke halaman petunjuk • Tombol pengaturan untuk masuk ke halaman pengaturan • Tombol kredit untuk masuk ke halaman kredit • Tombol keluar untuk keluar dari aplikasi.

Gambar 3.7 Antar Muka Halaman Utama

Shopping Time	
Halaman Petunjuk	
Visualisasi	**Keterangan**
  	• Background merupakan papan tulis. • Berisi petunjuk cara bermain *shopping time* yang terdiri dari: ▪ Uang jajan yang diberikan harus habis dalam waktu tertentu. ▪ Waktu yang diberikan untuk menyelesaikan misi yaitu menghabiskan uang jajan yang diberikan. ▪ Klik toko untuk membeli barang. ▪ Jawab pertanyaan yang ada lalu klik tombol beli**.** ▪ Lakukan lahkah 3 dan 4 sampai uang jajan habis. • Ada tombol kembali untuk kembali ke halaman menu utama

Gambar 3.8 Antar Muka Halaman Petunjuk

Shopping Time	
Halaman Petunjuk	
05	**Visualisasi**
	• Background merupakan papan tulis. • Berisi pengaturan musik dan efek suara yang ada di permainan *shopping time* • Ada tombol kembali untuk kembali ke halaman menu utama

Gambar 3.9 Antar Muka Halaman Pengaturan

Shopping Time	
Halaman Petunjuk	
Visualisasi	**Keterangan**
	• Background merupakan papan tulis. • Berisi pembuat permainan *shopping time.* • Ada tombol kembali untuk kembali ke halaman menu utama.

Gambar 3.10 Antar Muka Halaman Kredit

3.2.2 Analisis Karakter/Properti

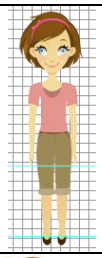
Dalam permainan *shopping time* terdapat dua karakter yaitu karakter pemain yang digambarkan sebagai anak perempuan dan karakter pencuri yang

digambarkan sebagai laki-laki yang memakai penutup mata dan menggunakan pakaian seperti pencuri. Selain itu, dalam permainan ini juga terdapat 6 toko yang menjual 4 barang yang berbeda dengan harga yang berbeda pula.

Berikut adalah gambaran properti dan karakter yang ada di permainan *shopping time*.

<table>
<tr><th colspan="2">Shopping Time</th></tr>
<tr><td colspan="2">Barang dan Karakter</td></tr>
<tr><th>Visualisasi</th><th>Keterangan</th></tr>
<tr><td>Toko:

Barang yang dijual:
</td><td>Toko dan barang yang dijual:
Toko mainan
Pesawat 1, harga Rp 15.000
Kapal layar1, harga Rp 20.000
Helikopter, harga Rp 22.500
Pesawat 2, harga Rp 25.000
Toko mobil-mobilan
Mobil 1, harga Rp 15.000
Mobil 2, harga Rp 20.000
Mobil 3, harga Rp 22.500
Mobil 4, harga Rp 25.000
Toko permen
Permen 1, harga Rp 10.000
Permen 2, harga Rp 15.000
Permen 3, harga Rp 17.500
Permen 4, harga Rp 20.000
Toko kue
Kue 1, harga Rp 5.000
Kue 2, harga Rp 7.500
Kue 3, harga Rp 10.000
Kue 4, harga Rp 12.500
Toko boneka
Boneka 1, harga Rp 15.000
Boneka 2, harga Rp 20.000
Boneka 3, harga Rp 25.000
Boneka 4, harga Rp 30.000
Toko ice cream
Ice cream 1, harga Rp 5.000
Ice cream 2, harga Rp 7.500
Ice cream 3, harga Rp 10.000
Ice cream 4, harga Rp 15.000</td></tr>
</table>

	Karakter orang/pemain: ▪ Nama: Kanza ▪ Kemampuan: sebagai karakter orang yang menjadi perwakilan pemain untuk menjalankan misi
	• Karakter musuh/pencuri ▪ Nama: Pencuri ▪ Kemampuan: bisa mencuri barang-barang yang ada di toko.

Gambar 3.11 Antar Muka Properti dan Karakter

3.2.3 *Storyboard*

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan *storyboard* penyampaian ide cerita bisa dilakukan dengan mudah. Hal ini disebabkan karena *storyboard* dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita.

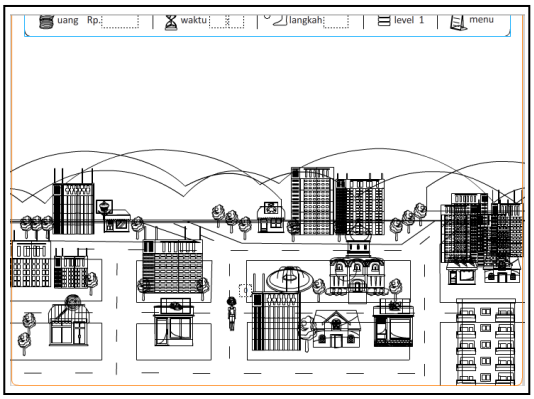
Berikut merupakan *storyboard* dari Aplikasi permainan *shopping time*:

1. Pemain memilih menu mulai



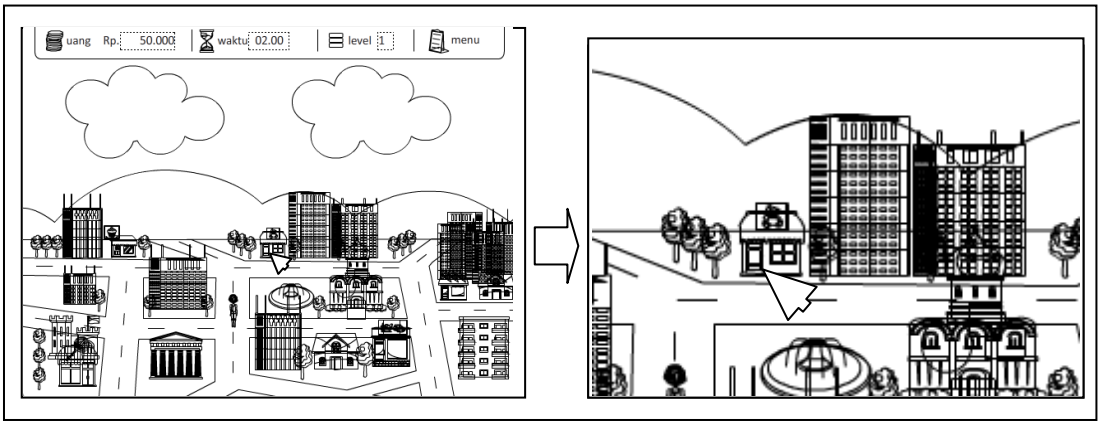
Gambar 3.12 *Storyboard* 1

2. Muncul halaman permainan yang berupa pusat perbelanjaan.



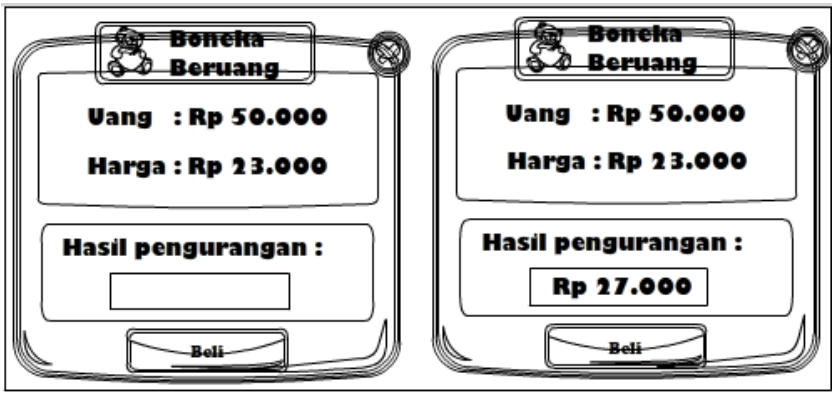
Gambar 3.13 *Storyboard* 2

3. Pemain memilih (meng-klik) salah satu toko untuk membeli barang.

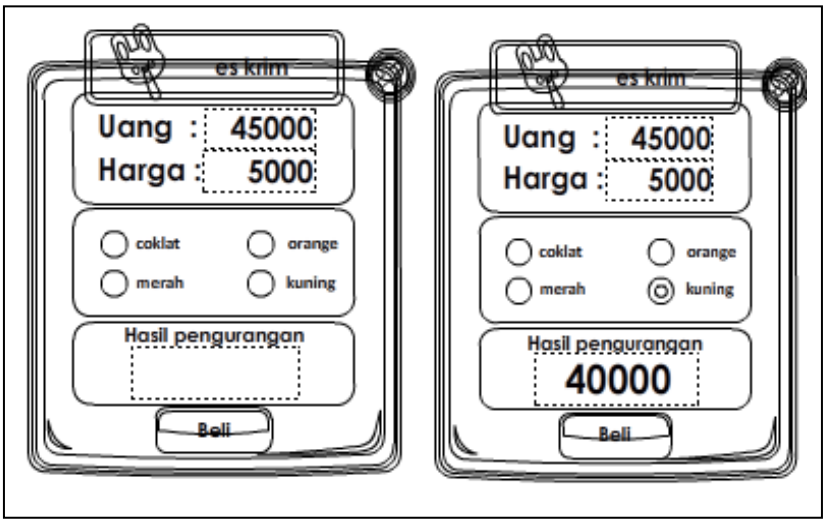


Gambar 3.14 *Storyboard* 3

4. Pemain menjawab pertanyaan yang ada

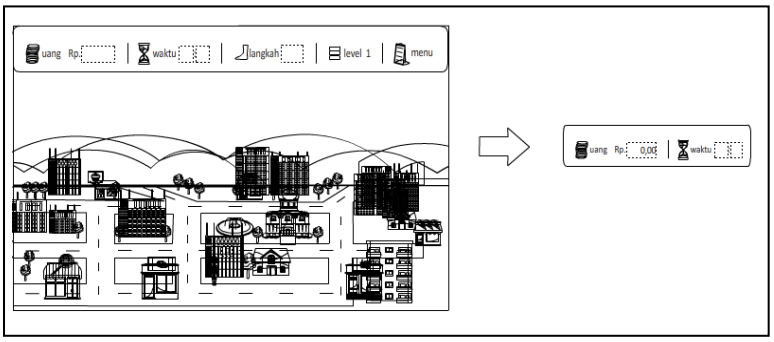


Gambar 3.15 *Storyboard* 4 level 1



Gambar 3.16 *Storyboard* 4 level 2 dan 3

5. Kembali lagi ke langkah 2-4 sampai uang jajan atau waktu habis



Gambar 3.17 *Storyboard* 5

6. Hati-hati dengan musih yang bisa mencuri barang yang telah dibeli.



Gambar 3.18 *Storyboard* 6

7. Jika misi selesai maka akan muncul halaman kemenangan lalu melanjutkan ke level berikutnya, jika tidak maka muncul halaman permainan berakhir.



Gambar 3.19 *Storyboard* 7

3.2.4 *Gameplay*

Permainan *shopping time* memiliki *gameplay* sebagai berikut:

- Permainan dimulai dengan layar yang menampilkan kawasan pusat perbelanjaan lengkap dengan toko-tokonya yang berjumlah 6 buah, dan dikatakan selesai jika pemain berhasil menghabiskan uang jajan yang diberikan dengan membelikan barang-barang yang ada dalam waktu yang telah ditentukan agar pemain mendapatkan *score* (nilai).
- Pemain harus menghabiskan uang tadi dengan membelikan barang-barang yang ada di toko-toko yang ada di pusat perbelanjaan.
- Saat pemain meng-klik toko, maka akan muncul kotak yang berisi nama barang dan harga barang.
- Untuk membeli barang tersebut, pemain harus bisa menjawab pertanyaan yang ada yaitu pengurangan antara uang yang ada dan harga barang dan di level dua dan tiga ada tambahan pertanyaan yaitu warna barang yang dibeli. Jika

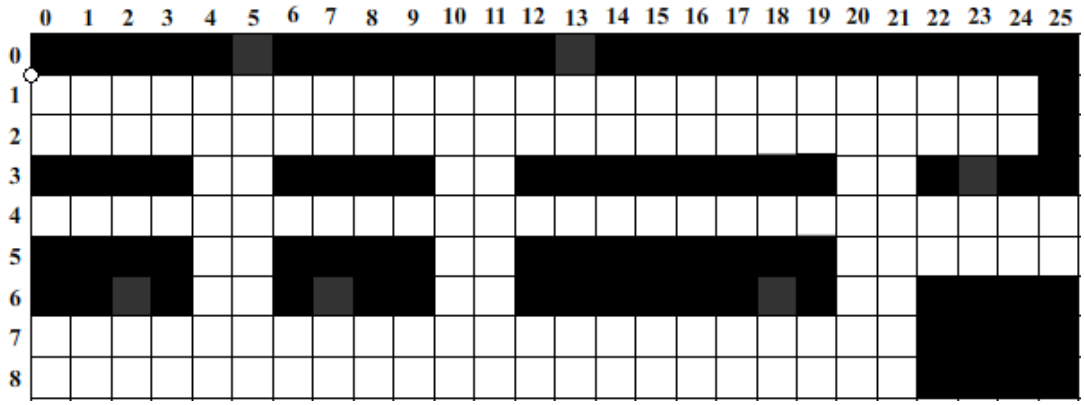
jawabannya benar maka saat tombol beli di klik maka akan muncul pesan "selamat, jawaban anda benar" lalu uang jajan langsung berkurang dan proses jaul beli selesai, untuk ke halaman awal permainan klik tombol map, jika jawaban tidak benar maka saat tombol beli di klik maka akan muncul pesan "jawaban salah, silahkan ulang lagi".

- Pada setiap toko hanya bisa dibeli satu barang saja.
- Permainan ini terdiri dari 3 level yakni:
  a. Level 1, waktu yang diberikan adalah 2 menit dan uang jajan yang diberikan adalah Rp 50.000. Pemain harus menjawab pertanyaan yaitu pengurangan antara uang jajan dengan harga barang yang dibeli.
  b. Level 2, waktu yang diberikan adalah 1,5 menit dan uang jajan yang diberikan adalah Rp 45.000. Pemain harus menjawab pertanyaan yaitu pengurangan antara uang jajan dengan harga barang yang dibeli dan apa warna barang yang dibeli.
  c. Level 3, waktu yang diberikan adalah 1 menit dan uang jajan yang diberikan adalah Rp 37.500. Pemain harus menjawab pertanyaan yaitu pengurangan antara uang jajan dengan harga barang yang dibeli dan apa warna barang yang dibeli dan ada NPC yang akan mencuri barang-barang yang ada di toko.
- Pemain hanya dapat memainkan level 3 jika telah berhasil melewati level 1 dan level 2.
- Permainan berakhir saat uang pemain habis atau waktu yang disediakan habis.

3.2.5 Analisis Penggunaan Algoritma A* dalam Permainan *Shopping Time*

Algoritma A* *(A Star)* akan diterapkan pada analisis kasus pencarian jalan terpendek menuju tujuan pada karakter pemain yang ada di permainan *shopping time* untuk mencari kecepatan, jarak langkah menuju tujuan dan simpul yang diperiksa.

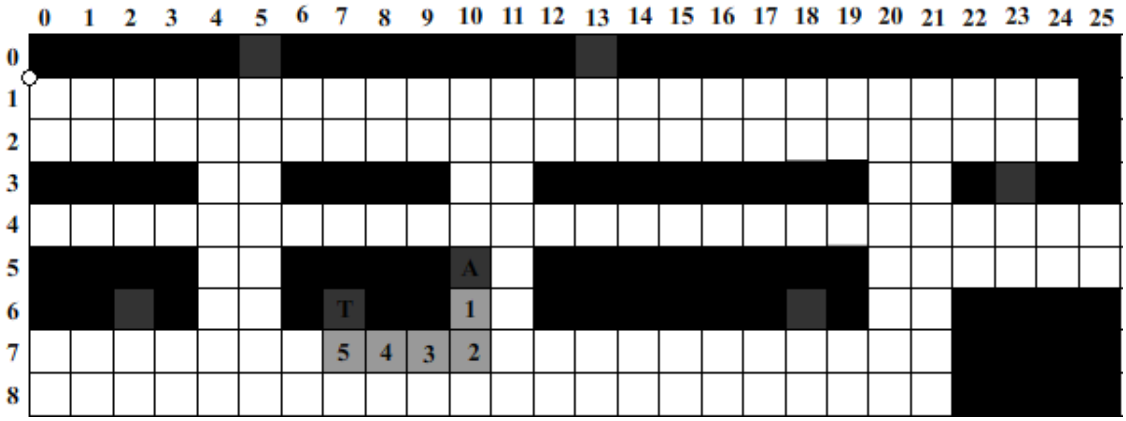
Dalam kondisi jarak yang ditempuh sangat jauh dan memiliki banyak rintangan algoritma A* cocok untuk mencari solusi yang terbaik. Setiap pergerakan yang dilakukan pada kotak statusnya akan disimpan pada suatu *list*. *List* ini akan digunakan untuk melakukan pengecekan apakah sudah pernah membangun status tersebut atau belum agar tidak menggerakkan kotak yang sama berkali-kali ke status yang sama. Dengan menerapkan strategi ini, selain menemukan solusi, algoritma ini juga bisa menemukan langkah terpendek untuk mencapai solusi tersebut. Gambar 3.7 merupakan peta permainan *shopping time*.



Gambar 3.20 Peta permainan *shopping time* 9x26

Pada gambar 3.7, kotak warna abu-abu adalah toko yanng meruakan *node* tujuan (ada 6 *node* tujuan). Kotak hitam adalah halangan yang tidak bisa dilewati.

Cara kerja dalam pencarian jalan pada permainan *shopping time* ini yakni setiap *node* akan memeriksa ke arah kiri, atas, kanan, dan bawahnya. Setiap *node* yang diperiksa dicari nilai f, nilai g, dan nilai h nya. *Node* dengan nilai f terendah yang dipilih dan dijadikan *parent node* untuk pencarian selanjutnya hingga *node* tujuan ditemukan. Contoh solusi pencarian jalan dapat terlihat pada gambar 3.8.



Gambar 3.21 Contoh solusi pencarian jalan permainan *shopping time*

Pada gambar 3.8, kotak A merupakan *node* awal, kotak T merupakan *node* tujuan dan kotak angka merupakan *rute* yang ditemukan. Maka jika diimplementasikan pada permainan *shopping time* suatu karakter akan melewati halangan dan akhirnya akan menemukan *rute* terpendek ke tujuan.

3.2.6 Analisis Algoritma A* pada Permainan *Shopping Time*

Terdapat beberapa hal yang perlu didefinisikan terlebih dahulu dalam kasus permainan *shopping time* dengan penerapan algoritma A* *(A Star)*. Adapun istilah-istilah yang akan dibahas yaitu *node*, *parent node*, *cost*, *open list*, *closed list*, nilai f, nilai g, nilai h dan heuristik.

1. *Node* (simpul): merupakan titik di sepanjang jalan. Dalam A* terdapat simpul awal dan simpul akhir. A* akan menghitung semua simpul dari awal sampai akhir untuk mendapatkan jalan terpendek.
2. *Parent node*: ketika sebuah simpul diuji dalam algoritma, simpul tersebut merupakan *parent node* untuk semua simpul yang terhubung langsung (simpul yang traversable dalam satu langkah). Dengan demikian, dapat dengan mudah bekerja mundur melalui semua *parent node* untuk membuat jalan setelah mencapai simpul tujuan.
3. *Cost* (biaya): biaya untuk setiap simpul didasarkan pada dua faktor yaitu biaya untuk sampai ke simpul tersebut dari simpul awal dan perkiraan biaya untuk mendapatkan dari simpul tersebut ke simpul akhir. Biaya *node* biasanya dinyatakan melalui 3 variabel f, g dan h. Simpul dengan biaya yang lebih rendah yang digunakan.
4. *Open List* adalah *list* yang menyimpan kemungkinan simpul yang akan diperiksa. *Open list* dibuat terurut berdasarkan nilai f. *open list* digunakan untuk menentukan secara selektif (berdasarkan nilai f) jalan yang dikira lebih dekat menuju pada simpul tujuan. *Open* berisi simpul-simpul yang masih memiliki peluang untuk terpilih sebagai simpul terbaik (*best node*).
5. *Closed List* adalah senarai (*list*) untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik (*best node*) atau senarai yang menyimpan jalan yang sudah diperiksa dari *open list*. Artinya, *closed list* berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup). Kedua *list* (*open list*

dan *closed list*) ini bertujuan juga untuk menghindari penelusuran berkali-kali jalan (*rute*) yang memang sudah diidentifikasi agar tidak masuk kembali ke dalam *open list*.

6. Nilai f adalah *cost* perkiraan suatu simpul yang teridentifikasi. Nilai F merupakan hasil dari f(n).
7. Nilai g hasil dari fungsi g(n), adalah banyaknya langkah yang diperlukan untuk menuju ke simpul sekarang.
8. Nilai h: merupakan estimasi harga simpul tersebut dihitung dari simpul tujuan. Nilai ini diperkirakan. Hal ini tidak tepat karena jalan yang akan ambil untuk sampai ke simpul akhir tidak diketahui. Oleh karena itu, biaya harus diperkirakan dengan menggunakan fungsi yang disebut heuristik.
9. Heuristik: merupakan fungsi untuk memperkirakan biaya dari simpul tertentu ke simpul akhir. Kelebihan A* adalah dapat mendukung berbagai fungsi heuristik, karena itu percobaan bisa dilakukan berbeda-beda sampai mendapatkan hasil yang dicari. Fungsi heuristik dapat berbeda berdasarkan kecepatan kerja, efisiensi dan berbagai kriteria lainnya.

Langkah-langkah yang dilakukan pada permainan *shopping time* menggunakan algoritma A* :

1. Tentukan *node* awal dan *node* tujuan
2. Tambahkan *node* awal ke *open list* (yang seharusnya kosong)
3. Pilih *node* saat ini dalam *open list* dengan biaya terendah
4. Jika *node* saat ini sama dengan tujuan, maka pencarian selesai dan lanjutkan ke langkah 8.

5. Periksa setiap *node* yang terhubung langsung, dengan cara:
    i. Jika *node* tidak traversable (tidak bisa bergerak ke sana karena *node* tersebut terpilih atau karena alasan lainnya) atau *node* sudah dalam *open list* atau dalam *close list*, maka dapat dilewati dan beralih ke *node* berikutnya dalam daftar *node* terhubung. Jika tidak, lanjutkan ke ii.
    ii. Menghitung biaya dari *node* tersebut (nilai f, nilai g, dan nilai h).
    iii. Mengatur *node* saat ini sebagai *parent node*.
    iv. Tambahkan *node* ke *parent node*.
6. Tambahkan *node* saat ini ke *close list*.
7. Kembali ke langkah 3-7 dengan *open list* yang diperbarui. (langkah ini harus tetap berjalan sampai *node* tujuan ditemukan)
8. *Node* tujuan telah ditemukan, buat daftar *node* yang akan menjadi daftar jalur dan tambahkan *node* tujuan ke daftar itu.
9. Tambahkan *parent node* dari *node* tujuan ke daftar jalur
10. Tambahkan *parent node* dari *node* sebelumnya ke daftar jalur, lakukan hal ini sampai ke *node* awal.

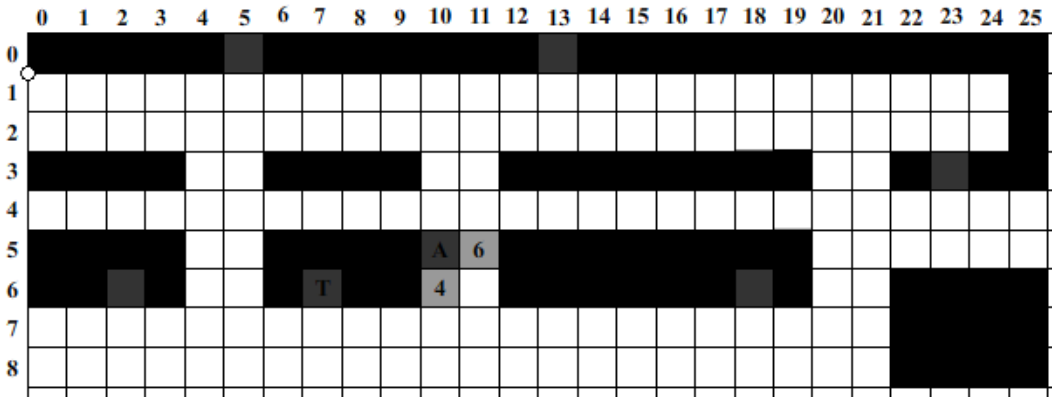
Berdasarkan peta permainan *shopping time* yang terdapat pada analisis kasus, maka perhitungan pemecahan masalah dengan algoritma A* untuk pencarian jalan dengan titik awal (10,5) dan titik tujuan (7,6) adalah sebagai berikut:

Posisi simpul awal = Ax : 10, Ay : 5

Posisi simpul tujuan = goal x : 7, goal y : 6

Tabel 3.4 Pencarian jalan dari awal (10,5) dan titik tujuan (7,6) langkah 1

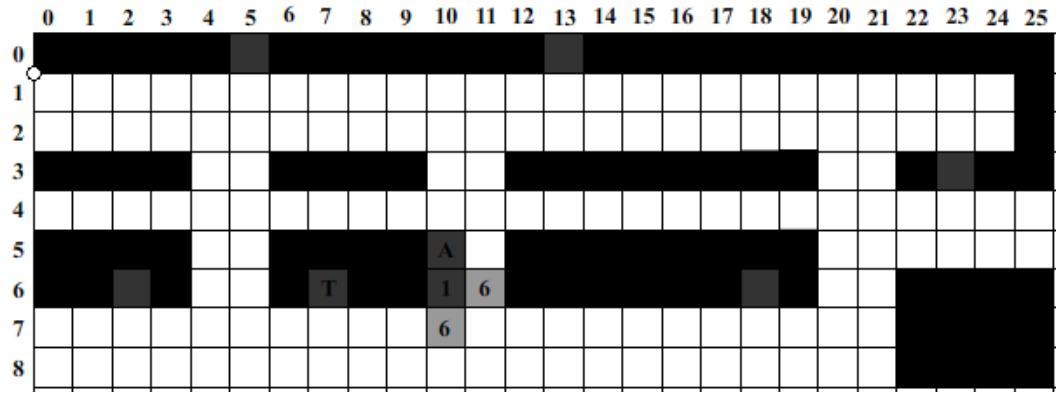
Langkah 1	
n (11,5)	g (11,5) = 1 h(n) = (abs(n.x-goal.x) + abs(n.y-goal.y)) h(11,5) = (abs(11 - 7) + abs(5 - 6)) = (abs(4) + abs(-1)) = 5 f (11,5) = g (11,5) + h (11,5) = 1 + 5 = 6
n (10,6)	g (10,6) = 1 h(n) = (abs(n.x-goal.x) + abs(n.y-goal.y)) h(10,6) = (abs(10 - 7) + abs(6 - 6)) = (abs(3) + abs(0)) = 3 f (10,6) = g (10,6) + h (10,6) = 1 + 3 = 4
n (9,5)	Merupakan penghalang.
Keterangan: Dari perhitungan diatas terdapat satu simpul yang mungkin menjadi *BestNode* yaitu (10,6) dengan f(n) = 4, maka dipilihlah simpul (10,6) dengan biaya yaitu 4.	



Gambar 3.22 Perhitungan Solusi A* Langkah 1

Tabel 3.5 Pencarian jalan dari awal (10,5) dan titik tujuan (7,6) langkah 2

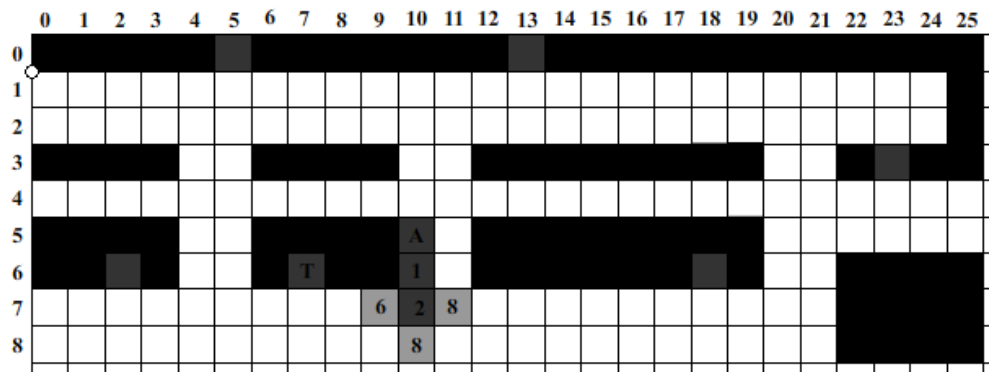
Langkah 2	
n (11,6)	g (11,6) = 2 h(n) = (abs(n.x-goal.x) + abs(n.y-goal.y)) h(11,6) = (abs(11 - 7) + abs(6 - 6)) = (abs(4) + abs(0)) = 4 f (11,6) = g (11,6) + h (11,6) = 2 + 4 = 6
n (10,7)	g (10,7) = 2 h(10,7) = (abs(10 - 7) + abs(7 - 6)) = (abs(3) + abs(1)) = 4 f (10,7) = g (10,7) + h (10,7) = 2 + 4 = 6
n (9,6)	Merupakan penghalang.
Keterangan: Dari perhitungan diatas terdapat dua simpul yang mungkin menjadi *BestNode* yaitu (11,6) dan (10,7) dengan f(n) = 6, maka dipilihlah simpul (10,7) dengan biaya yaitu 4 karena mendekati *node* tujuan.	



Gambar 3.23 Perhitungan Solusi A* Langkah 2

Tabel 3.6 Pencarian jalan dari awal (10,5) dan titik tujuan (7,6) langkah 3

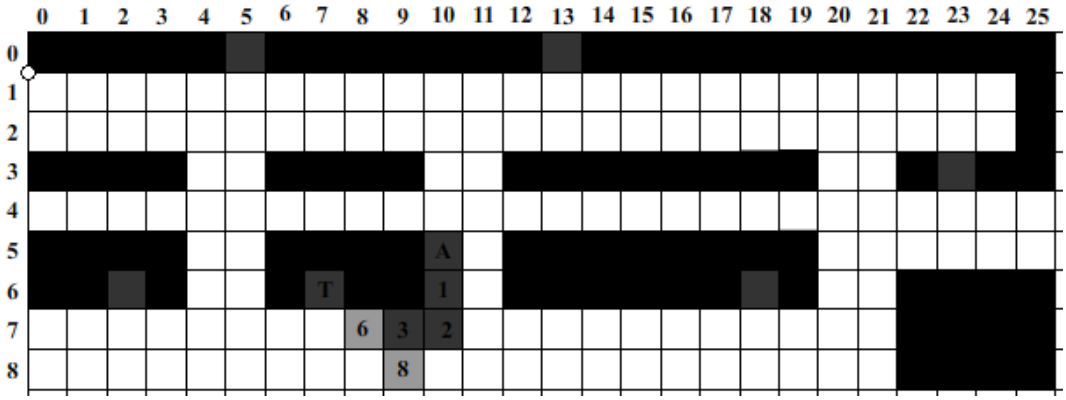
Langkah 3	
n (11,7)	g (11,7) = 3 h(11,7) = (abs(11 - 7) + abs(7 - 6)) = (abs(4) + abs(1)) = 5 f (11,7) = g (11,7) + h (11,7) = 3 + 5 = 8
n (10,8)	g (10,8) = 3 h(10,8) = (abs(10 - 7) + abs(8 - 6)) = (abs(3) + abs(2)) = 5 f (10,8) = g (10,8) + h (10,8) = 3 + 5 = 8
n (9,7)	g (9,7) = 3 h(9,7) = (abs(9 - 7) + abs(7 - 6)) = (abs(2) + abs(1)) = 3 f (9,7) = g (9,7) + h (9,7) = 3 + 3 = 6
Keterangan: Dari perhitungan diatas terdapat satu simpul yang mungkin menjadi *BestNode* yaitu (9,7) dengan f(n) = 6, maka dipilihlah simpul (9,7) dengan biaya yaitu 6.	



Gambar 3.24 Perhitungan Solusi A* Langkah 3

Tabel 3.7 Pencarian jalan dari awal (10,5) dan titik tujuan (7,6) langkah 4

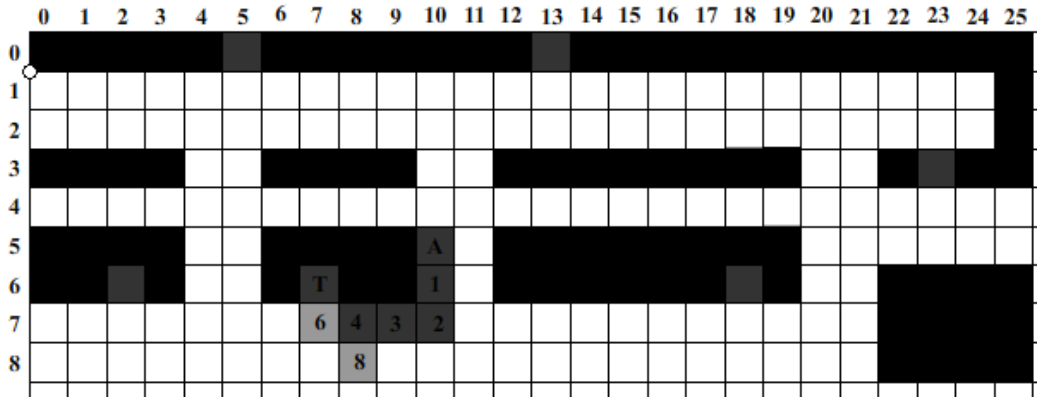
Langkah 4	
n (9,8)	g (9,8) = 4 h(9,8) = (abs(9 - 7) + abs(8 - 6)) = (abs(2) + abs(2)) = 4 f (9,8) = g (9,8) + h (9,8) = 4 + 4 = 8
n (8,7)	g (8,7) = 4 h(8,7) = (abs(8 - 7) + abs(7 - 6)) = (abs(1) + abs(1)) = 2 f (8,7) = g (8,7) + h (8,7) = 4 + 2 = 6
Keterangan: Dari perhitungan diatas terdapat satu simpul yang mungkin menjadi *BestNode* yaitu (8,7) dengan f(n) = 6, maka dipilihlah simpul (8,7) dengan biaya yaitu 6.	



Gambar 3.25 Perhitungan Solusi A* Langkah 4

Tabel 3.8 Pencarian jalan dari awal (10,5) dan titik tujuan (7,6) langkah 5

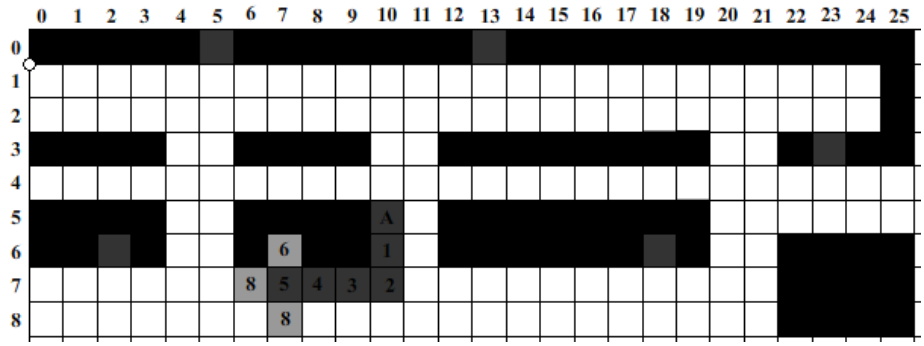
Langkah 5	
n (8,8)	g (8,8) = 5 h(8,8) = (abs(8 - 7) + abs(8 - 6)) = (abs(1) + abs(2)) = 3 f (8,8) = g (8,8) + h (8,8) = 5 + 3 = 8
n (7,7)	g (7,7) = 5 h(7,7) = (abs(7 - 7) + abs(7 - 6)) = (abs(0) + abs(1)) = 1 f (7,7) = g (7,7) + h (7,7) = 5 + 1 = 6
n (8,6)	Merupakan penghalang.
Keterangan: Dari perhitungan diatas terdapat satu simpul yang mungkin menjadi *BestNode* yaitu (7,7) dengan f(n) = 6, maka dipilihlah simpul (7,7) dengan biaya yaitu 6.	



Gambar 3.26 Perhitungan Solusi A* Langkah 5

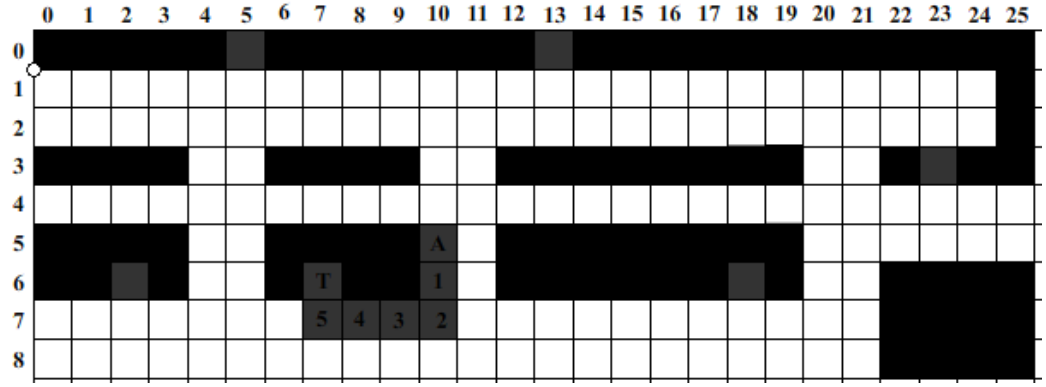
Tabel 3.9 Pencarian jalan dari awal (10,5) dan titik tujuan (7,6) langkah 6

Langkah 6	
n (7,8)	g (7,8) = 6 h(7,8) = (abs(7 - 7) + abs(8 - 6)) = (abs(0) + abs(1)) = 2 f (7,8) = g (7,8) + h (7,8) = 6 + 2 = 8
n (6,7)	g (6,7) = 6 h(6,7) = (abs(6 - 7) + abs(7 - 6)) = (abs(1) + abs(1)) = 2 f (7,7) = g (7,7) + h (7,7) = 6 + 2 = 8
n (7,6)	g (7,6) = 6 h(7,6) = (abs(7 - 7) + abs(6 - 6)) = (abs(0) + abs(0)) = 0 f (7,6) = g (7,6) + h (7,6) = 6 + 0 = 6
Keterangan: Dari perhitungan diatas terdapat satu simpul yang mungkin menjadi *BestNode* yaitu (7,6) dengan f(n) = 6 yang diketahui sebagai *node* tujuan, maka *node* tujuan ditemukan dan pencarian selesai.	



Gambar 3.27 Perhitungan Solusi A* Langkah 6

Dari perhitungan diatas, maka langkah atau jalan yang dilalui dari titik awal (10,5) sampai ke titik tujuan (7,6) adalah (10,6) – (10,7) – (9,7) – (8,7) – (7,7) yang diperlihatkan pada gambar berikut ini.



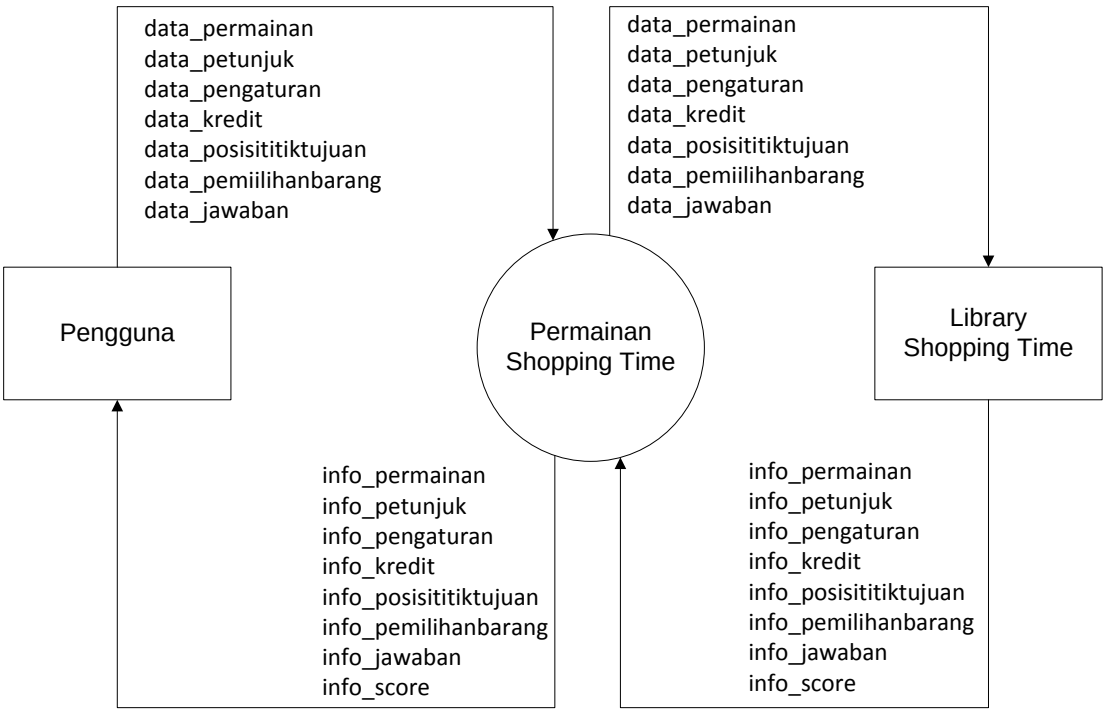
Gambar 3.28 Hasil pencarian jalan dengan Algoritma A*

3.3 Analisis Kebutuhan Fungsional

Pemodelan yang digunakan pada aplikasi yang akan dibuat adalah aliran data terstuktur sebagai alat bantu yaitu diagram konteks dan DFD (*Data Flow Diagram*).

3.3.1 Diagram Konteks

Diagram konteks atau disebut juga dengan model aplikasi fundamental yang merepresentasikan seluruh elemen aplikasi sebagai sebuah *bubble* tunggal dengan data masukan-keluaran yang ditunjukkan oleh anak panah yang masuk dan keluar secara berurutan. Diagram konteks dari aplikasi yang akan dibangun adalah seperti gambar 3.16.



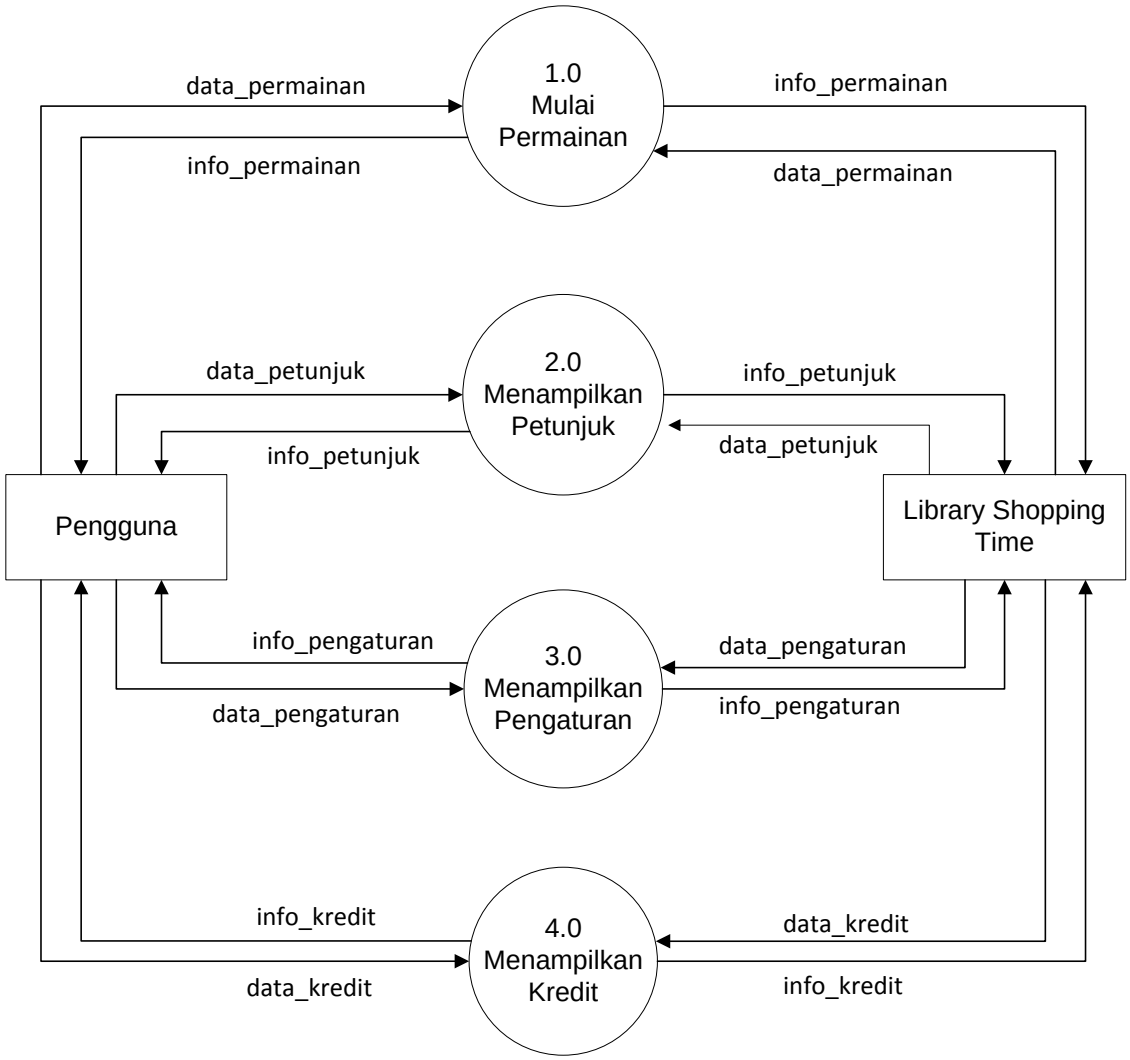
Gambar 3.29 Diagram Konteks Pada Permainan *Shopping Time*

3.3.2 *Data Flow* Diagram (DFD)

Data flow diagram merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan. Salah satu keuntungan DFD adalah memudahkan pemakai atau *user* yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan.

3.3.2.1 DFD Level 0

DFD Level 0 dibuat jika pada diagram konteks masih terdapat proses yang harus dijelaskan lebih rinci. Pada DFD Level 0 terdapat proses - proses yaitu pengaturan titik tujuan, proses pencarian *rute* dan pembelian barang yang dapat dilihat pada gambar 3.17.



Gambar 3.30 DFD Level 0 Permainan *Shopping Time*

1. Pengaturan Mulai Permain

   Proses pembelian barang-barang yang tersedia.

2. Menampilkan Petunjuk

   Proses menampilkan halaman petunjuk.

3. Menampilkan Pengaturan

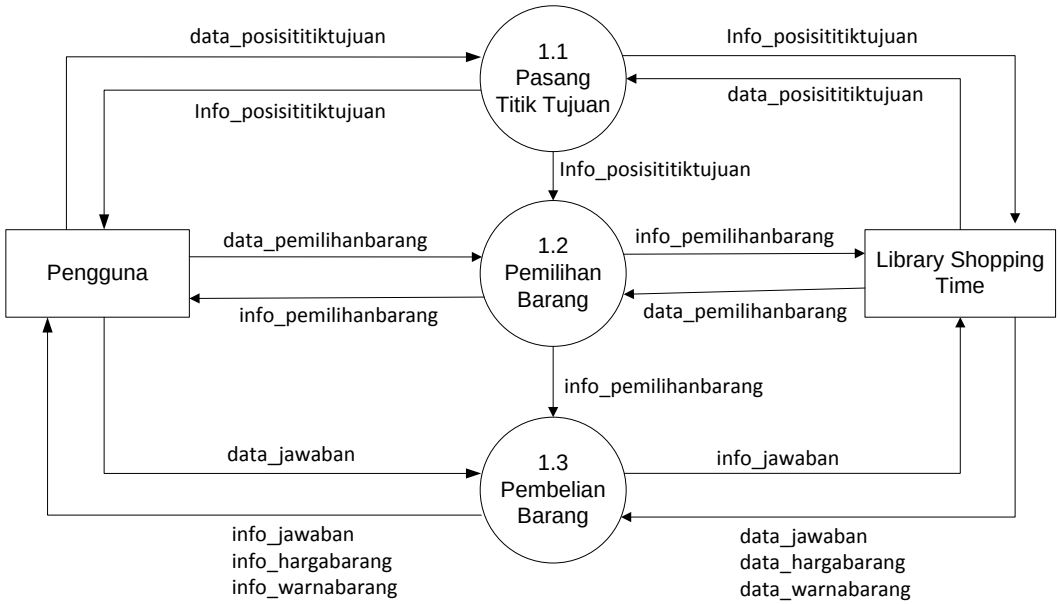
   Proses menampilkan halaman pengaturan.

4. Menampilkan Kredit

   Proses menampilkan halaman kredit.

3.3.2.2 DFD Level 1 Proses 1.0 Mulai Permainan

Proses yang terdapat pada DFD level 1 proses 1.0 adalah proses mulai permainan yang terdiri atas proses 1.1 pasang titik tujuan, proses 1.2 pilih barang, proses 1.3 pembelian barang. DFD level 1 untuk proses 1.0 dapat dilihat pada gambar 3.18.



Gambar 3.31 DFD Level 1 Proses 1.0 Mulai Permainan

1.1 Pasang Titik Tujuan

Proses untuk pemasangan titik tujuan pada *map*.

1.2 Pilih Barang

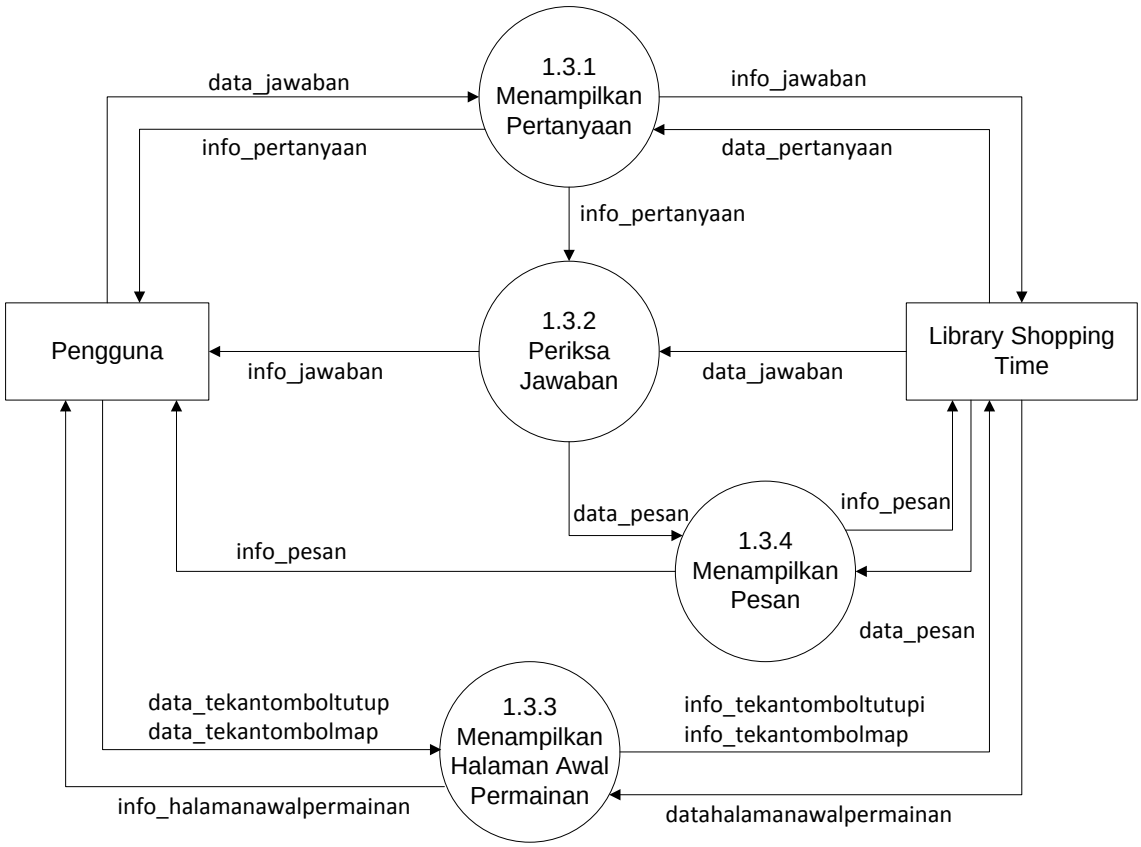
Proses untuk memilih barang yang akan dibeli

1.3 Pembelian Barang

Proses pembelian barang yang diinginkan

3.3.2.3 DFD Level 2 Proses 1.3 Pembelian Barang

Proses yang terdapat pada DFD level 2 proses 1.3 adalah proses pengaturan titik tujuan yang terdiri atas proses 1.3.1 jawab pengurangan, proses 1.3.2 jawab warna barang, proses 1.3.3 tekan tombol beli, proses 1.3.4 tekan tombol tutup, proses 1.3.5 tekan tombol map, proses 1.3.6 tampil pesan. DFD level 2 untuk proses 1.3 dapat dilihat pada gambar 3.19.



Gambar 3.32 DFD Level 2 Proses 1.3 Pembelian Barang

1.3.1 Menampilkan Pertanyaan

Proses untuk menampilkan pertanyaan yang ada.

1.3.2 Periksa Jawaban

Proses untuk memeriksa jawaban yang diberikan pemain.

1.3.3 Menampilkan Halaman Awal Permainan

Proses untuk menampilkan halaman awal permainan saat pemain telah selesai melakukan pembelian barang atau pun saat pemain membapalkan pembelian barang.

1.3.4 Menampilkan Pesan

Proses menampilkan pesan jawaban benar atau salah saat menjawab pertanyaan.

3.3.2.4 Spesifikasi Proses

Untuk menjelaskan lebih lanjut tentang proses-proses yang ada di dalam diagram alir data atau DFD maka dibuatlah spesifikasi proses. Spesifikasi proses merupakan deskripsi dari setiap elemen proses yang terdapat dalam aplikasi, yang meliputi nama proses, masukan, keluaran dan keterangan dari proses. Adapun spesifikasi proses untuk diagram alir data permainan *shopping time* dapat dilihat pada tabel 3.10.

Tabel 3.10. Spesifikasi proses

No.	Proses	Keterangan
1.	No. Proses	1.0
	Nama Proses	Memulai permainan
	Source (sumber)	Pengguna
	Deskripsi	Proses memulai permainan *shoping time*.
	Masukan	Data mulai permainan.
	Keluaran	Info halaman permainan.
	Logika proses	1. Pengguna menekan tombol mulai 2. Aplikasi memunculkan halaman permainan
2.	No. Proses	2.0
	Nama Proses	Menampilkan Petunjuk
	Source (sumber)	Pengguna
	Deskripsi	Proses menampilkan halaman petunjuk cara bermain *shopping time*
	Masukan	Data petunjuk cara bermain *shopping time*
	Keluaran	Info petunjuk cara bermain *shopping time*
	Logika proses	1. Pengguna menekan tombol petunjuk 2. Aplikasi memunculkan halaman petunjuk permainan
3	No. Proses	3.0
	Nama Proses	Menampilkan Pengaturan
	Source (sumber)	Pengguna
	Deskripsi	Proses menampilkan halaman pengaturan
	Masukan	Data pengaturan permainan *shopping time*
	Keluaran	Info pengaturan permainan *shopping time*
	Logika proses	1. Pengguna menekan tombol pengaturan 2. Aplikasi memunculkan halaman pengaturan permainan

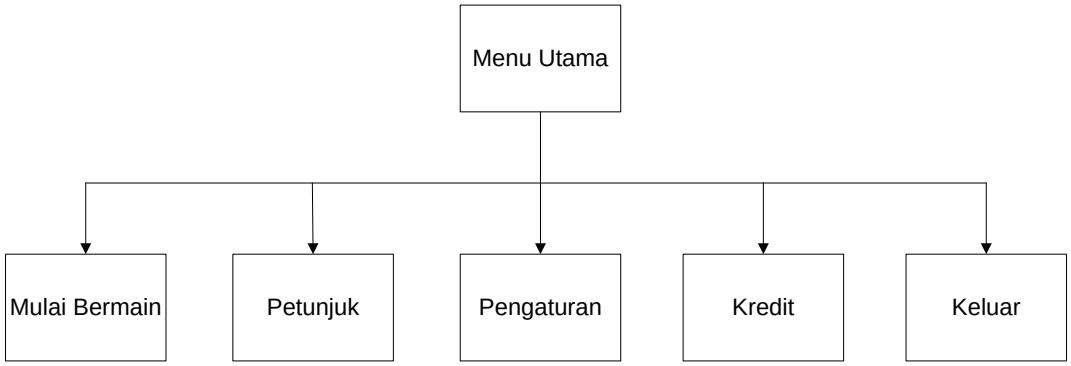
4	No. Proses	4.0
	Nama Proses	Menampilkan Kredit
	Source (sumber)	Pengguna
	Deskripsi	Proses menampilkan halaman kredit
	Masukan	Data kredit permainan *shopping time*
	Keluaran	Info kredit permainan *shopping time*
	Logika proses	1. Pengguna menekan tombol kredit 2. Aplikasi memunculkan halaman kredit permainan
5	No. Proses	1.1
	Nama Proses	Pasang titik tujuan
	Source (sumber)	Pengguna
	Deskripsi	Proses menempatkan titik tujuan pada *map*.
	Masukan	Data posisi titik tujuan
	Keluaran	Info posisi titik tujuan
	Logika proses	1. Algoritma A* berjaman menemukan jalan optimal. 2. Karakter bergerak sesuai dengan jalan yang telah ditemukan algoritma A*.
6	No. Proses	1.2
	Nama Proses	Pilih Barang
	Source (sumber)	Pengguna
	Deskripsi	Proses memilih barang yang akan dibeli.
	Masukan	Data barang
	Keluaran	Info barang
	Logika proses	1. Aplikasi memunculkan halaman toko yang menampilkan barang-barang yang dijual. 2. Pengguna memilih barang yang akan dibeli.
7	No. Proses	1.3
	Nama Proses	Pembelian Barang
	Source (sumber)	Pengguna
	Deskripsi	Proses pembelian barang.
	Masukan	Data harga barang, warna barang dan jawaba pertanyaan
	Keluaran	Info harga barang, warna barang dan hasil pembelian barang
	Logika proses	1. Pengguna menjawab pertanyaan yang ada (untuk level 1 pengurangan uang dengan harga barand, dilevel 2 pertanyaan ditambah dengan warna barang yang dibeli). 2. Pengguna menekan tombol beli.

		3. Aplikasi menampilkan uang pengguna yang telah berkurang.
8	No. Proses	1.3.1
	Nama Proses	Menampilkan Pertanyaan
	Source (sumber)	Pengguna
	Deskripsi	Proses menampilkan pertanyaan yang ada
	Masukan	Data jawaban dan data pertanyaan
	Keluaran	Info pertanyaan
	Logika proses	1. Aplikasi menampilkan kotak pertanyaan. 2. Pengguna mengisi *textbox* dengan jawaban pengurangan antara uang dan harga barang dan mengisi *radio button* untuk menjawab warna barang yang dibeli.
9	No. Proses	1.3.2
	Nama Proses	Periksa Jawaban
	Source (sumber)	Pengguna
	Deskripsi	Proses memeriksa jawaban yang diberikan pemain.
	Masukan	Data jawaban
	Keluaran	Info jawaban
	Logika proses	1. Pengguna menekan tombol beli setelah menjawab pertanyaan yang ada. 2. Aplikasi memeriksa jawaban yang diberikan pemain.
10	No. Proses	1.3.3
	Nama Proses	Menampilkan Halaman Awal Permainan
	Source (sumber)	Pengguna
	Deskripsi	Proses menampilkan halaman awal permainan saat proses belanja selesai atau proses belanja dibatalkan.
	Masukan	Data tombol tutup dan tombol map.
	Keluaran	Info halaman awal permainan.
	Logika proses	1. Pengguna menekan tombol tutup atau tombol map. 2. Aplikasi menampilkan halaman awal permainan.
11	No. Proses	1.3.4
	Nama Proses	Tampil Pesan
	Source (sumber)	Pengguns
	Deskripsi	Proses menampilkan pesan jawaban benar atau salah.
	Masukan	Data pesan.
	Keluaran	Info pesan.
	Logika proses	1. Pengguna menekan tombol beli.

		2. Jika jawaban benar, maka muncu pesan “jawaban benar” dan jika jawaban salah maka akan muncul pesan “jawaban salah, silahkan ulangi lagi”.

3.3.3 Perancangan Struktur Menu

Perancangan struktur menu berisikan menu dan submenu yang berfungsi memudahkan *user* didalam menggunakan sistem. Pada gambar 3.20 dijelaskan perancangan struktur menu permainan.



Gambar 3.33 Struktur Menu

3.4 Perancangan arsitektur

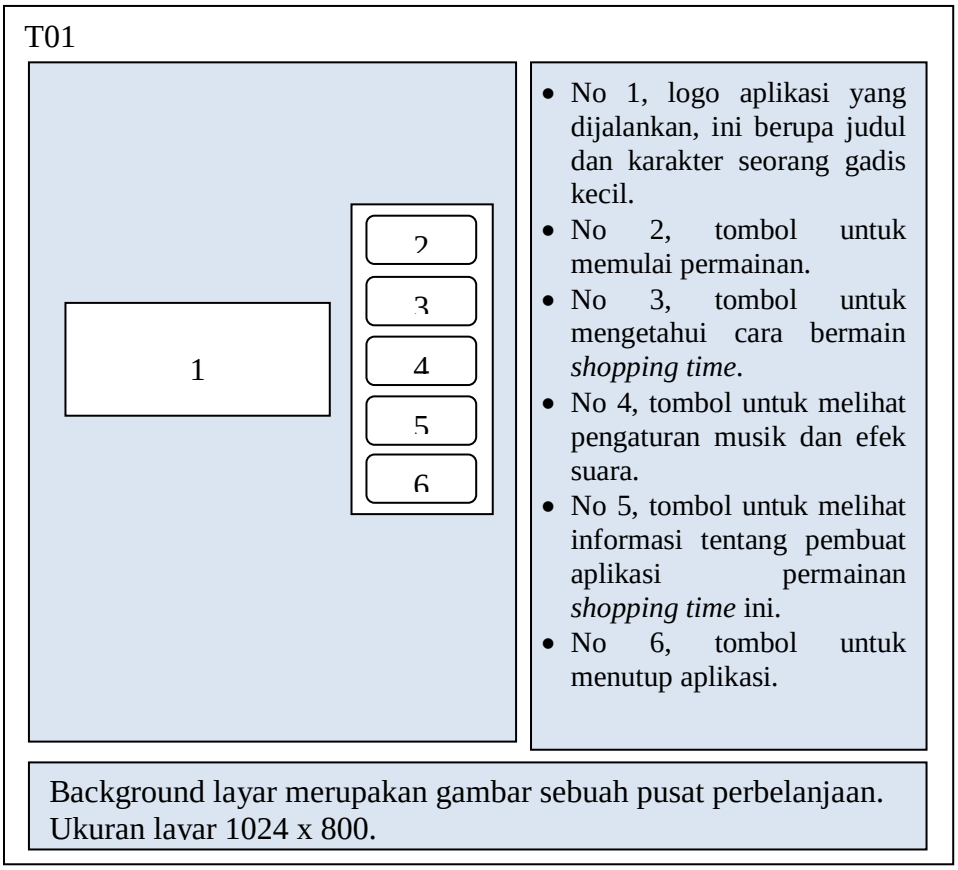
Perancangan arsitektur merupakan perancangan yang dibuat sebelum permainan dibangun. Dengan perancangan arsitektur akan mempermudah proses pembangunan aplikasi permainan *shopping time*.

3.4.1 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi.

3.4.1.1 *Form* Menu Utama

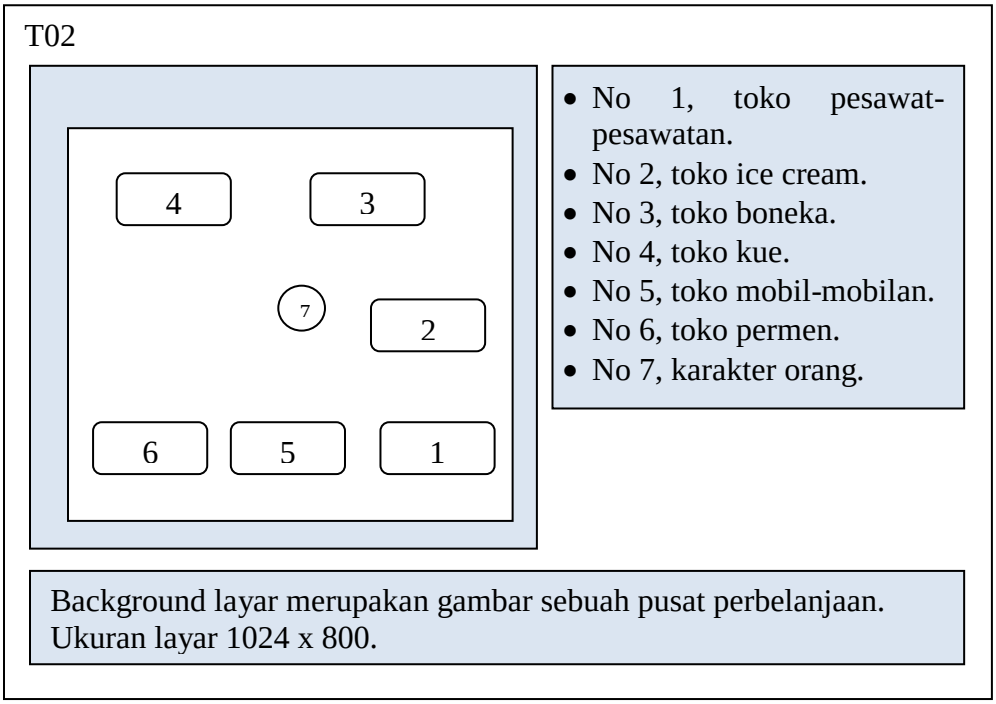
Form ini merupakan tampilan pertama saat aplikasi *shopping time* dijalankan. Pada rancangan *form* ini terdapat tombol-tombol yang mempunyai fungsi tertentu yang berguna untuk menjalankan aplikasi. Rancangan *form* menu utama dapat dilihat pada gambar berikut:



Gambar 3.34 Rancangan *Form* Menu Utama

3.4.1.2 *Form* Permainan

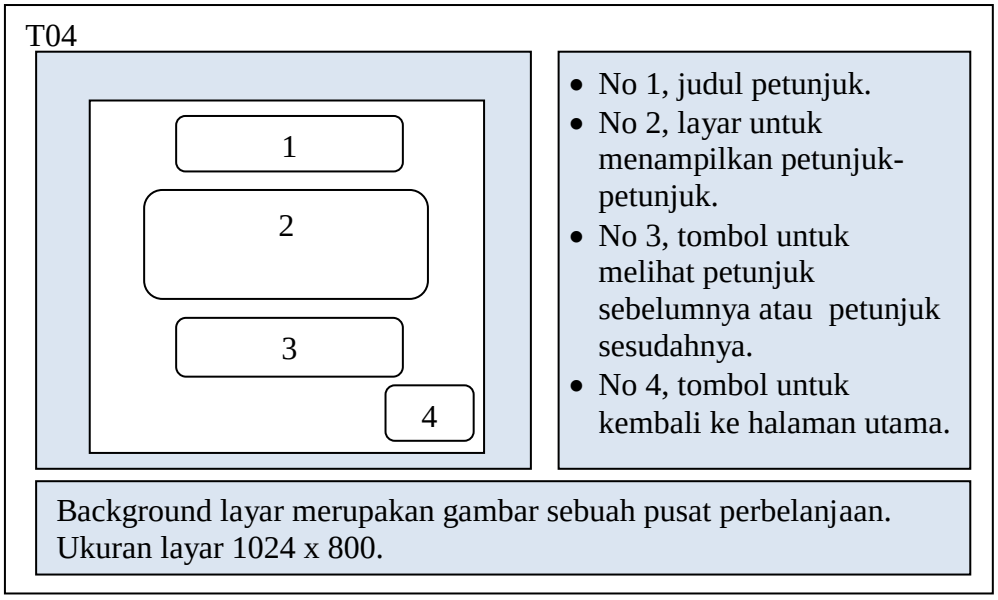
Form permainan merupakan rancangan *form* inti dari aplikasi *shopping time*. Di *form* inilah pemain akan memainkan aplikasi *shopping time*. Perancangan tampilan *form* ini dapat dilihat pada gambar berikut:



Gambar 3.35 Rancangan *Form* Permainan

3.4.1.3 *Form* Petunjuk

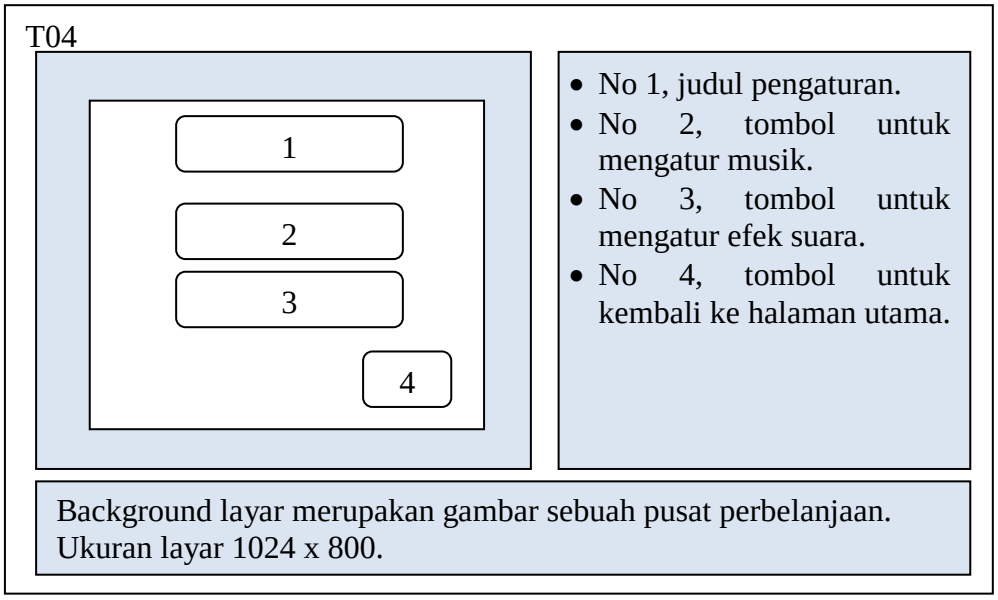
Form petunjuk merupakan informasi tentang bagaimana cara memainkan permainan *shopping time*. Adapun rancangan *form* ini adalah sebagai berikut:



Gambar 3.36 Rancangan *Form* Petunjuk

3.4.1.4 *Form* Pengaturan

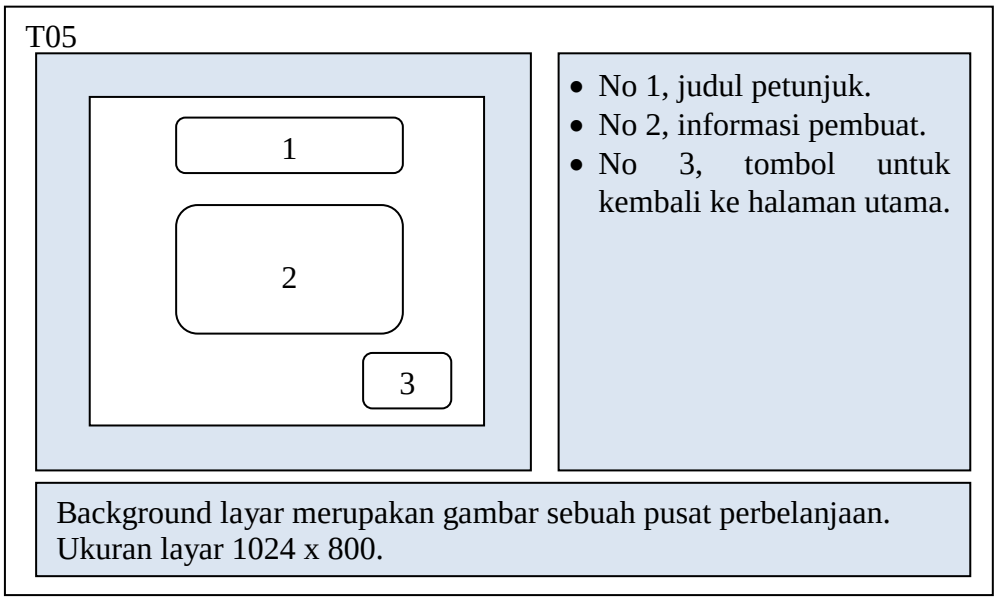
Form pengaturan merupakan informasi tentang pengaturan musik dan efek suara yang ada di permainan *shopping time*. Adapun rancangan *form* ini adalah sebagai berikut:



Gambar 3.37 Rancangan *Form* Pengaturan

3.4.1.5 *Form* Kredit

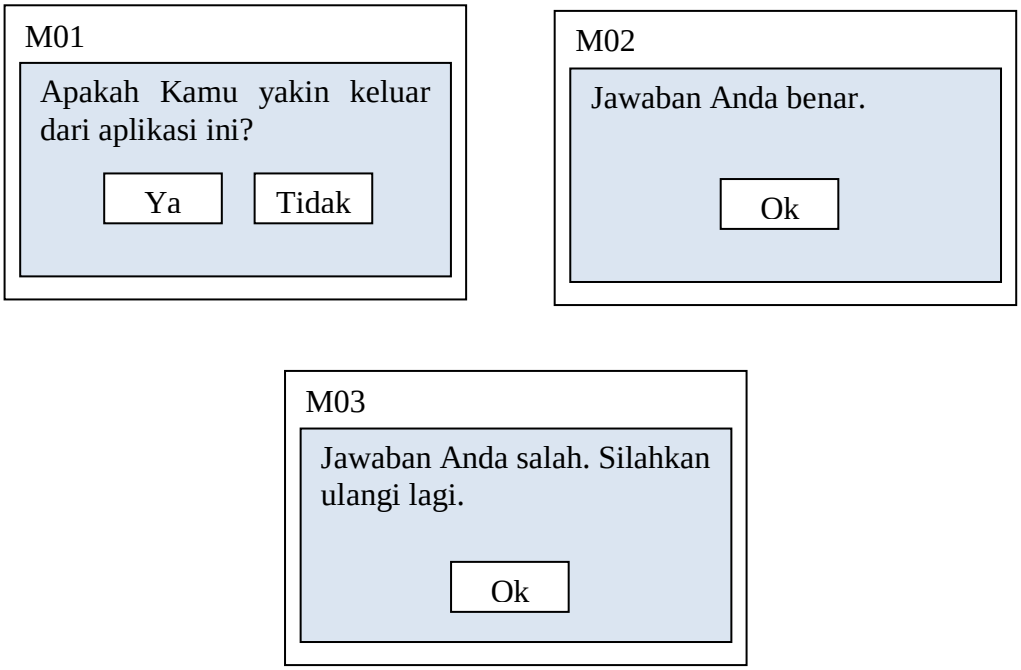
Form kredit merupakan informasi tentang pembuat permainan *shopping time*. Adapun rancangan *form* ini adalah sebagai berikut:



Gambar 3.38 Rancangan *Form* Kredit

3.4.1.6 Rancangan tampilan pesan

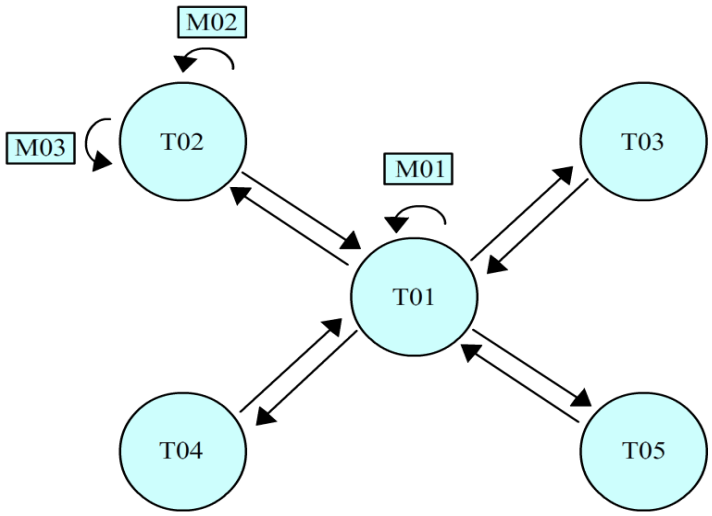
Perancangan Antar Muka ini adalah tampilan pesan yang muncul pada aplikasi ini seperti ditunjukan pada gambar.



Gambar 3.39 Rancangan Antarmuka Pesan

3.4.2 Jaringan Semantik

Perancangan jaringan semantik menggambarkan alur informasi dari setiap antarmuka aplikasi yang dijelaskan pada gambar berikut:



Gambar 3.40 Jaringan semantik

3.4.3 Perancangan Sistem

Perancangan sistem merupakan metodologi pengembangan suatu perangkat lunak yang dilakukan setelah melalui tahapan analisis. Perancangan bertujuan untuk memberikan gambaran secara terperinci dan merupakan tahap lanjutan dari analisis sistem. Dimana dalam hal ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengodean ke dalam suatu bahasa pemrograman. Desain umum yang akan diaplikasikan bertujuan untuk memberikan gambaran secara umum kepada penggunaan tentang sistem yang akan dibangun.

3.4.3.1 Perancangan Komponen Permainan *Shopping Time*

Perancangan komponen pada permainan *shopping time* terdiri dari pemain, titik awal dan titik tujuan, penghalang dan *map* yang akan dijelaskan sebagai berikut :

1. Pemain

Permainan ini akan dimainkan oleh satu orang. Pemain dapat menempatkan titik tujuan pada *map* yang telah disediakan dengan posisi yang diinginkan.

2. Titik awal dan titik tujuan

Titik awal merupakan titik pertama jalan yang akan dicari sedangkan titik tujuan merupakan titik akhir yang akan ditemukan. Titik awal akan digambarkan dengan gambar seorang gadis dan titik tujuan dengan gambar toko supaya tampilan lebih menarik.

3. Penghalang

Penghalang yaitu suatu titik yang tidak dapat dilewati. Penghalang yang akan digunakan yaitu gedung dan bangunan lain agar tampilan yang dihasilkan lebih menarik.

4. Map

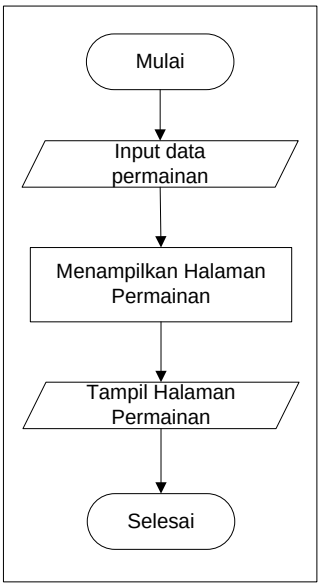
Map yang digunakan pada permainan *shopping time* ini akan berupa kotak-kotak dengan ordo 9 x 26.

3.4.4 Perancangan Prosedural

Sebagai langkah terakhir dalam perancangan yaitu perancangan prosedural yang akan di implementasikan ke dalam sistem. Prosedur ini akan digunakan sebagai algoritma dasar dalam mengkodekan prosedur yang ada.

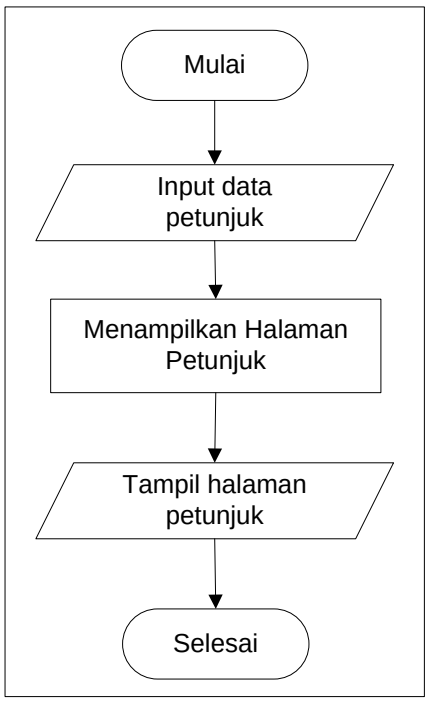
Adapun perancangan prosedural untuk pembangunan aplikasi permainan *shopping time* adalah sebagai berikut.

1. Prosedur mulai permainan *shopping time*



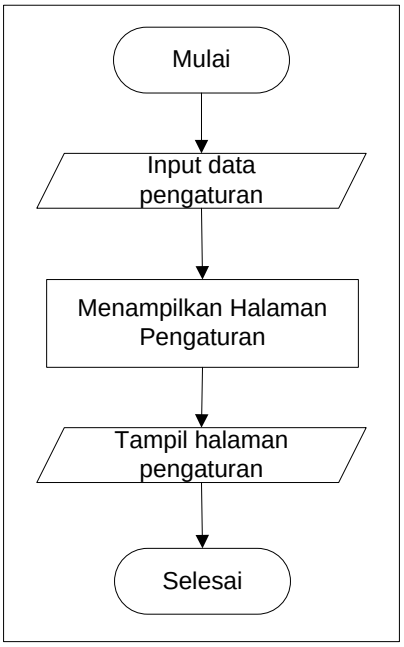
Gambar 3.41 *Flowchart* Mulai Permain *Shopping Time*

2. Prosedur petunjuk permainan *shopping time*



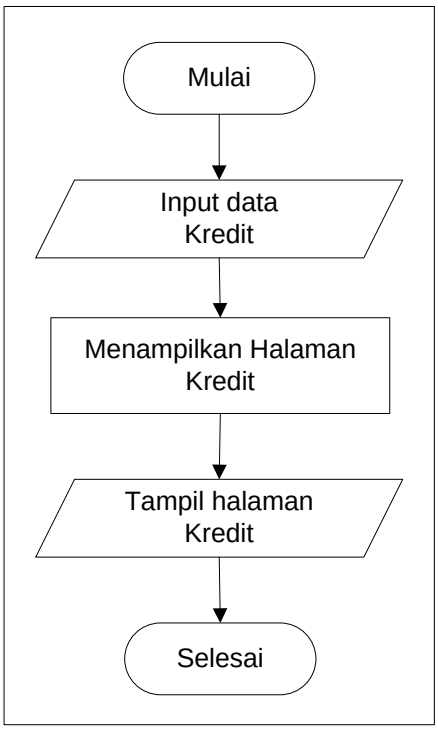
Gambar 3.42 *Flowchart* Petunjuk *Shopping Time*

3. Prosedur pengaturan permainan *shopping time*



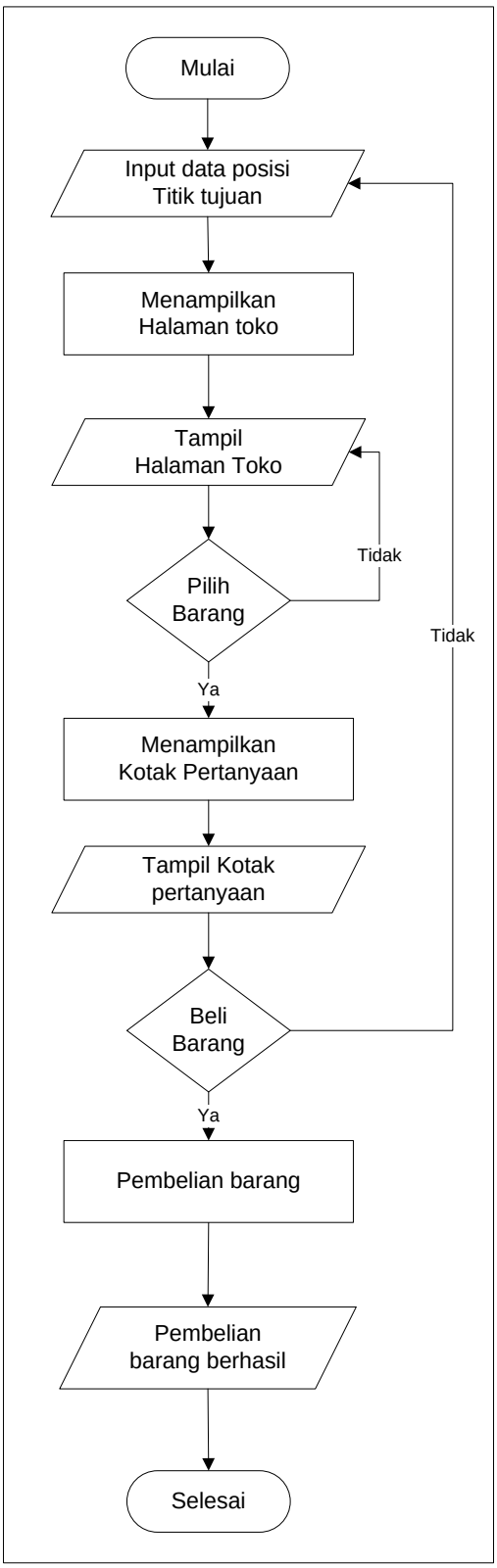
Gambar 3.43 *Flowchart* Pengaturan *Shopping Time*

4. Prosedur kredit permainan *shopping time*



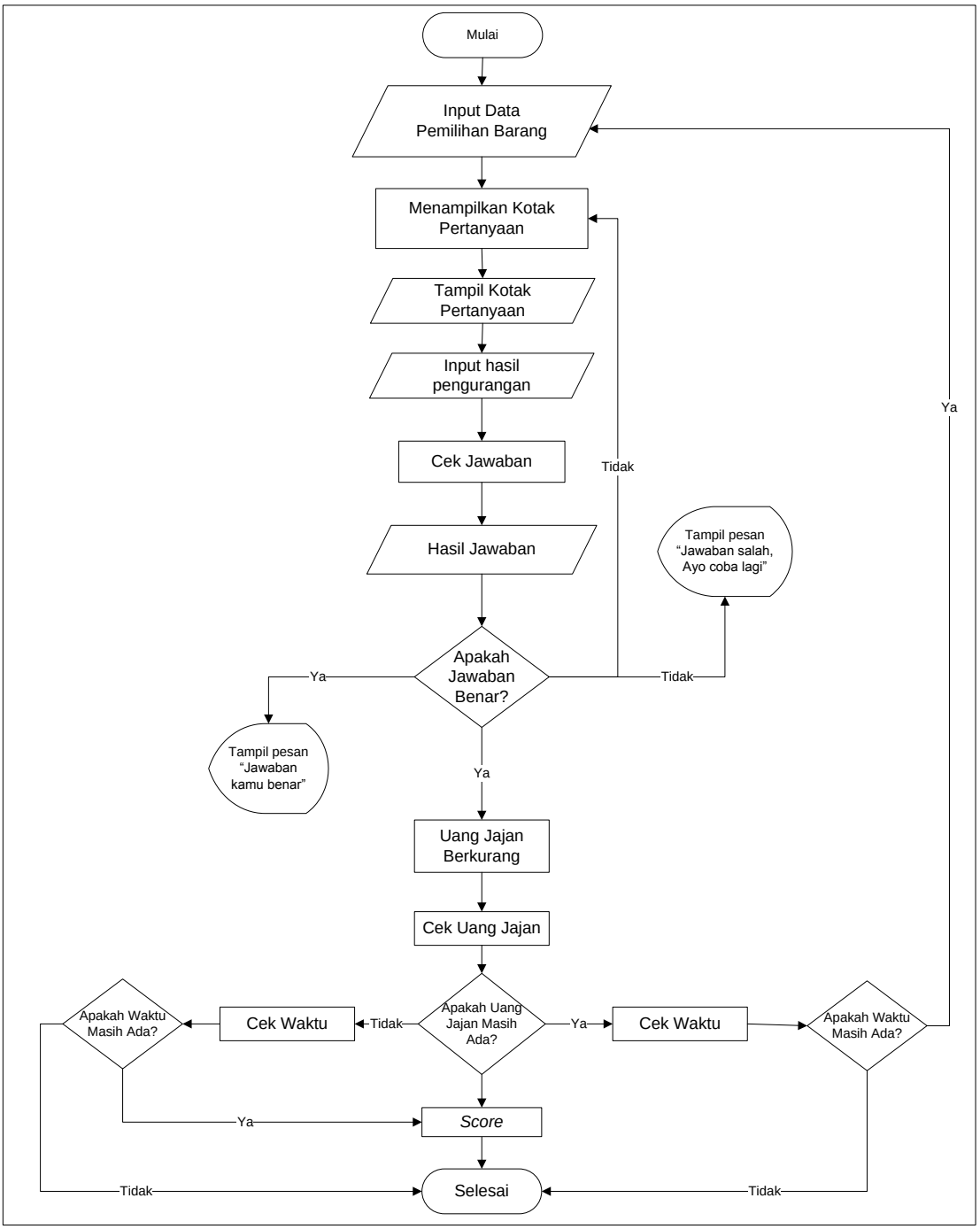
Gambar 3.44 *Flowchart* Kredit *Shopping Time*

5. Prosedur mulai bermain *shopping time*



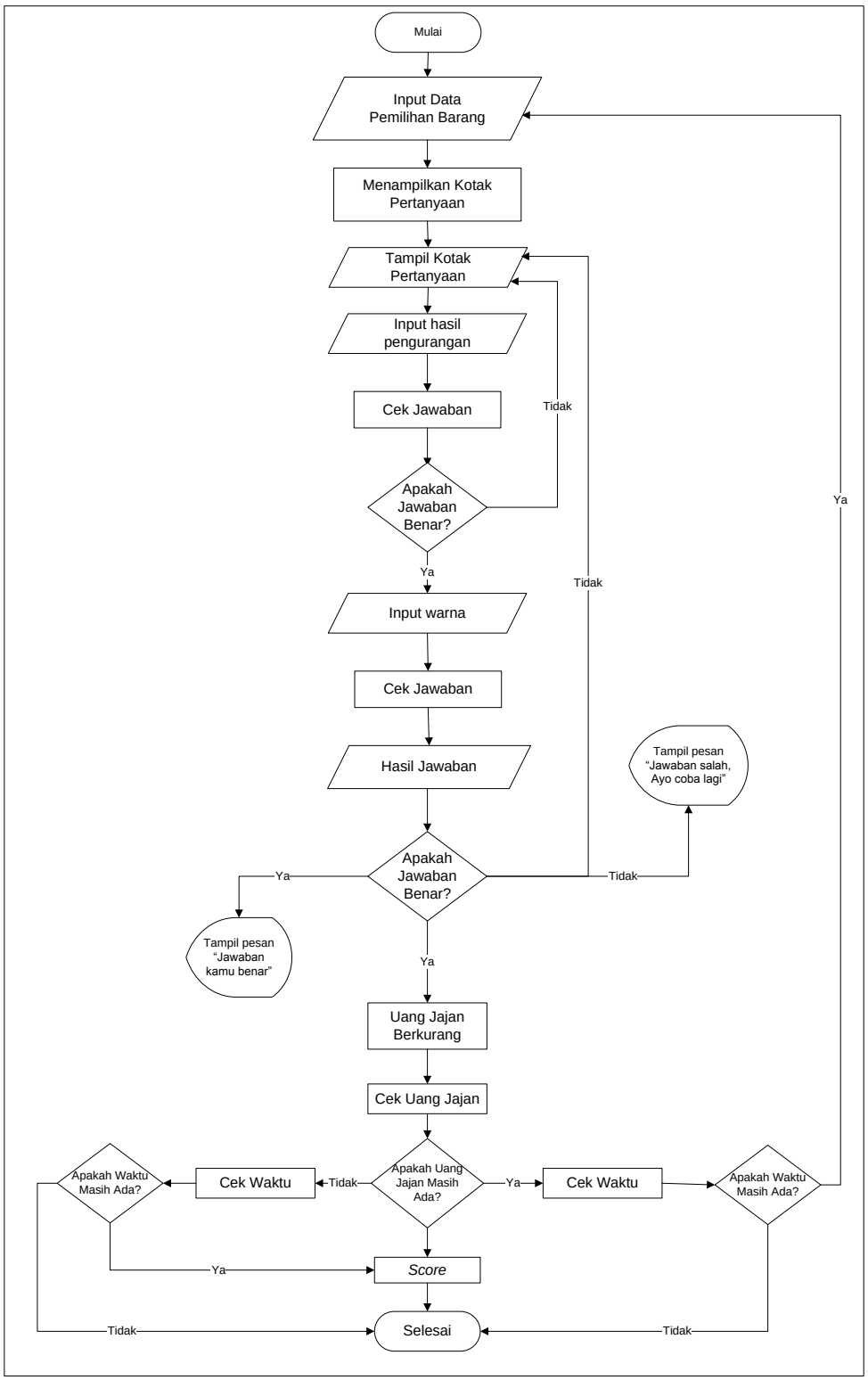
Gambar 3.45 *Flowchart* Mulai Bermain *Shopping Time*

6. Prosedur pembelian barang level 1



Gambar 3.46 *Flowchart* Pembelian Barang Level 1

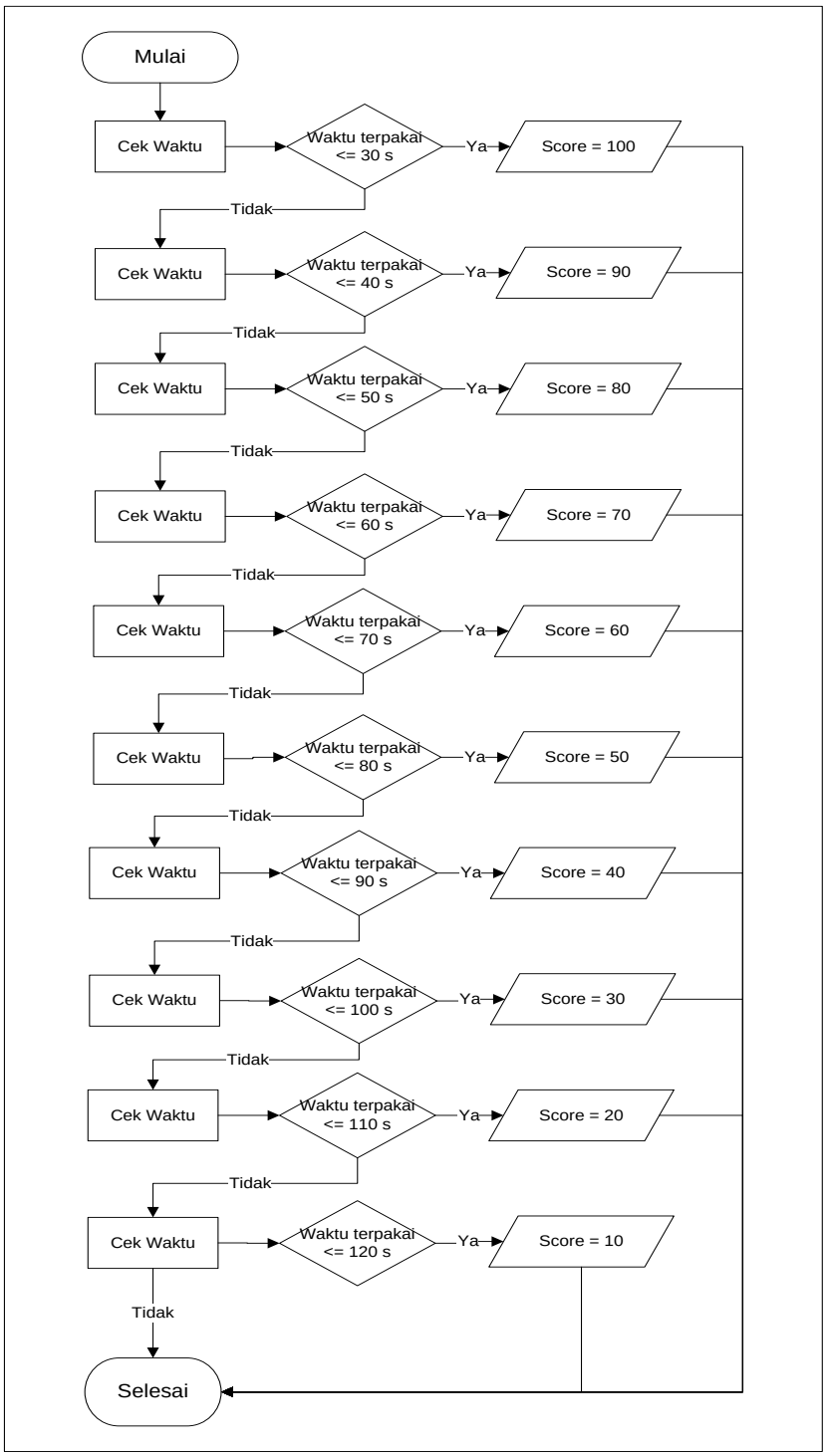
7. Prosedur pembelian barang Level 2



Gambar 3.47 *Flowchart* Pembelian Barang Level 2 dan 3

8. Prosedur *score*

   Setelah *user* menyelesaikan misi, maka akan muncul *score* yang diperoleh. Diagram alir *score* dapat dilihat pada gambar 3.21.



Gambar 3.48 *Flowchart Score*