Harvest Walukow
164231104

Latihan 1

1. Mengubah statement ke dalam kalkulus predikat

Ronaldo adalah seorang mahasiswa	$Mahasiswa(Ronaldo)$
Ronaldo masuk jurusan teknologi sains data	$Masuk(Ronaldo,TSD)$
Setiap mahasiswa teknologi sains data pasti mahasiswa teknik	$\forall x(MahasiswaTSD(x) \rightarrow MahasiswaTeknik(x))$
Kalkulus adalah mata kuliah yang sulit	$MataKuliahSulit(Kalkulus)$
Setiap mahasiswa teknik pasti akan suka kalkulus atau akan membencinya	$\forall x(MahasiswaTeknik(x) \rightarrow (Suka(x,Kalkulus) \vee Benci(x,Kalkulus)))$
Setiap mahasiswa pasti akan suka terhadap suatu mata kuliah	$\forall x \exists y(Mahasiswa(x) \rightarrow Suka(x,y))$
Mahasiswa yang tidak pernah hadir pada mata kuliah sulit, maka mereka pasti tidak suka terhadap mata kuliah tersebut	$\forall x \forall y((Mahasiswa(x) \wedge MataKuliahSulit(y) \wedge \neg Hadir(x,y)) \rightarrow \neg Suka(x,y))$
Mahasiswa yang tidak pernah hadir pada mata kuliah sulit, maka mereka pasti tidak suka terhadap mata kuliah tersebut	$\neg Hadir(Ronaldo,Kalkulus)$

Latihan 2

1. Langkah-langkah yang digunakan agen:
    - Knowledge Base (KB)
      Agen menyimpan informasi dalam bentuk proposisi yang merepresentasikan kondisi dunia.
        - Jika ada bau di suatu sel, maka Wumpus ada di salah satu sel tetangga.

- Jika ada angin di suatu sel, maka ada lubang di salah satu sel tetangga.
- Jika ada kilauan, maka gold ada di sel tersebut.

- Inferensi dengan Aturan Logika
  Agen menggunakan aturan logika seperti modus ponens untuk menyimpulkan keberadaan objek berbahaya atau gold.
  - Breeze(x,y) → (Pit(x-1,y) ∨ Pit(x+1,y) ∨ Pit(x,y-1) ∨ Pit(x,y+1))(Jika ada angin di (x,y), maka ada lubang di salah satu sel sekitarnya.)
  - Stench(x,y) → (Wumpus(x-1,y) ∨ Wumpus(x+1,y) ∨ Wumpus(x,y-1) ∨ Wumpus(x,y+1))(Jika ada bau di (x,y), maka ada Wumpus di salah satu sel tetangga.)

- Pengambilan Keputusan
  - Agen menggunakan informasi yang dikumpulkan untuk menentukan langkah selanjutnya.
  - Jika agen yakin bahwa suatu sel aman berdasarkan inferensi dari KB, maka ia akan bergerak ke sana.
  - Jika agen menemukan kilauan (glitter), ia akan mengambil gold.

2. Breadth-First Search (BFS)
   - Buat antrian kosong (queue) untuk menyimpan jalur eksplorasi.
   - Tambahkan posisi awal agen ke dalam antrian sebagai titik awal pencarian.
   - Simpan daftar sel yang sudah dikunjungi untuk menghindari eksplorasi ulang.
   - Selama antrian tidak kosong, lakukan langkah berikut (loop):
     a. Ambil posisi terdepan dari antrian.
     b. Periksa apakah posisi tersebut adalah gold. Jika ya, hentikan pencarian.
     c. Periksa apakah posisi tersebut aman (tidak mengandung Wumpus atau lubang) menggunakan aturan logika proposisional.
   - Jika posisi aman, tambahkan semua tetangga yang belum dikunjungi ke dalam antrian.
   - Gunakan aturan berikut untuk memvalidasi tetangga sebelum ditambahkan:
     - Jika ada bau, berarti ada kemungkinan Wumpus di salah satu sel tetangga.
     - Jika ada angin, berarti ada kemungkinan lubang di salah satu sel tetangga.
     - Hanya tambahkan sel yang tidak pasti berbahaya ke dalam antrian.
   - Ulangi langkah eksplorasi sampai:
     - Gold ditemukan
     - Tidak ada lagi sel aman yang bisa dieksplorasi (agen menyerah).
   - Jika gold ditemukan, agen akan mengikuti jalur terpendek kembali ke titik awal.

- Jika gold tidak ditemukan, agen akan kembali ke posisi awal dan keluar dari dunia Wumpus.

3. Agen ghost dalam Pacman dapat mencari pemain dengan munggunakan algoritma A* Search. Berikut detailnya:
    - Masukkan posisi awal Ghost ke dalam open list.
    - Tetapkan g-cost awal = 0.
    - Hitung h-cost ke Pacman.
    - Atur f(n) = g(n) + h(n) untuk node awal.
    - Selama open list tidak kosong, lakukan:
      a. Ambil node dengan f(n) terkecil dari open list.
      b. Jika node tersebut adalah posisi Pacman → Selesai!
      c. Jika bukan, pindahkan ke closed list dan lanjutkan ke langkah berikutnya.
    - Periksa empat arah pergerakan (atas, bawah, kiri, kanan).
    - Jika node tetangga bukan dinding dan belum dikunjungi:
        - Hitung g(n) baru (biaya langkah dari Ghost ke node baru).
        - Hitung h(n) menggunakan jarak Manhattan.
        - Hitung f(n) = g(n) + h(n).
        - Jika f(n)f(n)f(n) lebih kecil dari nilai sebelumnya di node tersebut, perbarui jalur.
    - Tambahkan semua node yang valid ke open list dan pilih yang memiliki f(n) terkecil.
    - Ulangi proses hingga Pacman ditemukan atau tidak ada jalur tersisa.