Soal 1

Reinforcement learning adalah salah satu sub-bidang dalam machine learning dimana pembelajaran dilakukan oleh suatu agent dengan cara berinteraksi dengan lingkungannya (environment). Tujuan utama dari agent adalah memaksimalkan reward yang didapat selama ia berinteraksi dengan lingkungannya. Alur sederhana dari suatu proses dalam reinforcement learning adalah sebagai berikut:

1. Agent berada pada suatu state tertentu
2. Agent mengambil action yang bisa dilakukan pada state tersebut
3. Lingkungan akan memberi reward berdasarkan action yang telah dilakukan oleh agent
4. Agent berpindah state
5. Ulangi poin 1 sampai 4 sebanyak n iterasi

Soal 2

Secara umum, Q-Learning akan melakukan iterasi secara terus-menerus selama sekian iterasi untuk mengupdate Q-Table. Q-Table merupakan tabel yang digunakan sebagai acuan agent dalam mengambil action dalam sebuah state. Berikut alur dari algoritma Q-learning:

1. Inisialisasi Q-table dengan nilai 0 dengan ukuran $banyak\_state \times banyak\_action$
2. Pilih action A yang mungkin pada state S menggunakan suatu policy tertentu

   Policy yang digunakan pada implementasi program ini adalah epsilon greedy policy. Terdapat dua pilihan action pada epsilon greedy policy, pilihan pertama adalah memilih action yang memiliki nilai Q maksimal pada state tertentu, pilihan kedua adalah memilih action secara random. Policy tersebut akan memilih aturan pemilihan berdasarkan probabilitas sebesar *epsilon*. Misalkan epsilon = 0.7, maka probabilitas terpilihnya aturan pemilihan action menggunakan nilai Q adalah sebesar 0.7.

   Policy ini dipilih untuk menyeimbangkan antara *exploration* dan *exploitation*. Exploration adalah kegiatan agent melakukan eksplorasi terhadap environment yang ada, sementara exploitation adalah kegiatan agent mengambil action yang paling menguntungkan menurutnya untuk saat itu. Kedua hal tersebut harus diseimbangkan supaya agent tidak terjebak dalam suatu local extremum.

3. Tambahkan reward terhadap reward kumulatif
4. Update Q-table sesuai dengan persamaan berikut:

$$Q(S_t, A_t) := Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right)$$

S_t: State pada waktu ke-t
A_t: Action pada waktu ke-t
$Q(S_t, A_t)$: Nilai Q (dari Q-table) pada S_t dan A_t
α: Learning rate
R_{t+1}: Reward pada waktu ke-t+1
γ: Discount rate
$\max_a Q(S_{t+1}, a)$: Nilai Q maksimum pada state S_{t+1}

5. Ubah state menjadi state yang baru
6. Ulangi poin 2-5 sampai mencapai terminal state
7. Ketika sudah mencapai terminal state, ulangi dari poin 2 dengan environment dan state agent yang direset ke posisi semula (memasuki episode baru).
8. Ulangi poin 2-7 sampai mencapai batas episode maksimal.

Soal 3

Q-Learning merupakan algoritma RL off-policy, sementara SARSA merupakan algoritma RL on-policy. Perbedaannya terletak pada policy yang digunakan pada saat mengubah nilai Q. Q-learning akan memilih nilai Q optimal pada state selanjutnya apapun policy yang digunakan saat itu. Sementara SARSA akan mengestimasi nilai Q berdasarkan policy yang diikutinya sekarang. Berikut merupakan persamaan untuk memperbarui nilai Q berdasarkan SARSA:

$$Q(S_t, A_t) := Q(S_t, A_t) + \alpha\big(R_{t+1} + Q(S_{t+1}, a_{t+1}) - Q(S_t, A_t)\big)$$

Q-learning akan mengerucut ke solusi optimal secara lebih cepat dan efisien dibandingkan dengan SARSA karena Q-learning dapat memanfaatkan policy lain (policy yang berbeda pada saat ini) untuk mengestimasi Q-value pada state berikutnya. Namun untuk permasalahan yang membatasi agent untuk hanya memanfaatkan policy saat ini, performanya menjadi kurang optimal. Untuk permasalahan yang demikian, SARSA akan memiliki performa yang lebih baik. Salah satu contoh permasalahan yang bersifat demikian adalah navigasi robot pada suatu lingkungan tertentu yang belum diketahui.