**PRAKTIKUM**

**REKAYASA PERANGKAT LUNAK**

**0953224026**

Haryono Setiadi, S.T., M.Eng.

**USE CASE DIAGRAM**

****

Disusun Oleh :

Muhammad Nur Hikmah Ramadhan

M0521054

Kelompok 3 – Sistem Booking DNA SPOT Cafe

Kelas B

**PROGRAM STUDI S-1 INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA**

**UNIVERSITAS SEBELAS MARET**

**SURAKARTA**

**2023**

1. **Percobaan *Use Case* Diagram untuk *Vending Machine***

Terdapat sebuah mesin penjual minuman kaleng otomatis. Mesin tersebut menjual 5 macam minuman dengan harga sama yaitu Rp. 5.000,-. Pembeli dapat menggunakan pecahan uang kertas Rp. 1.000,- hingga 5.000,-, namun harus sesuai jumlahnya dengan harga (tidak ada kembalian). Setiap 14 hari sekali, operator akan datang dan mengganti minuman dengan stok yang baru, serta mengambil uang yang telah terkumpul pada mesin untuk disetorkan kepada perusahaan penjual.

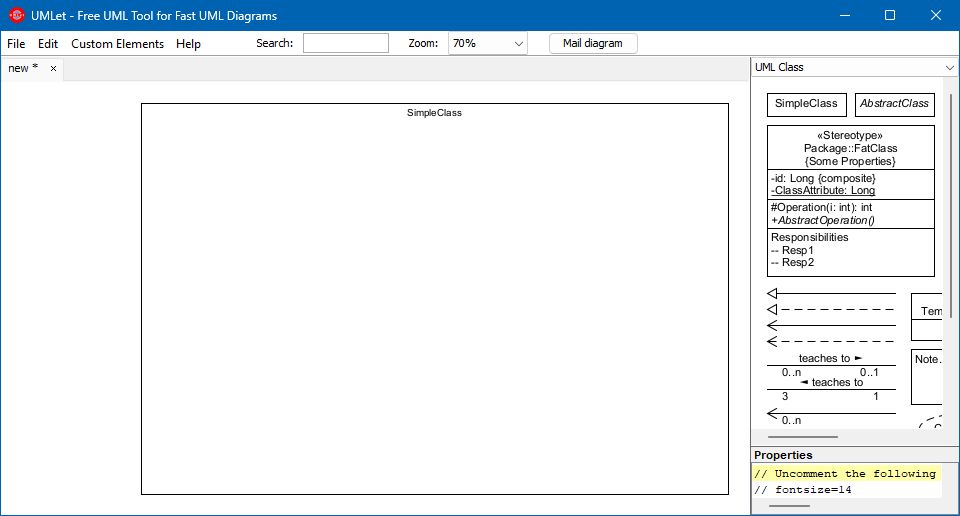
Buatlah sebuah use case diagram dari kasus vending machine tersebut!

**Jawaban**

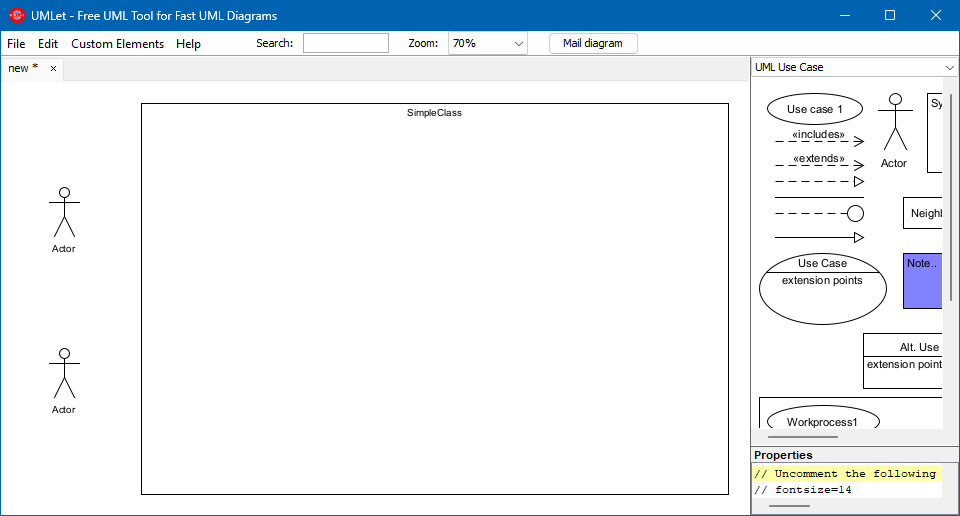
Identifikasikan aktor dengan kata ganti orang atau entitas eksternal sistem. Dalam kasus ini, digunakan kata ganti **Pembeli** dan **Operator**.

**Langkah-langkah** :

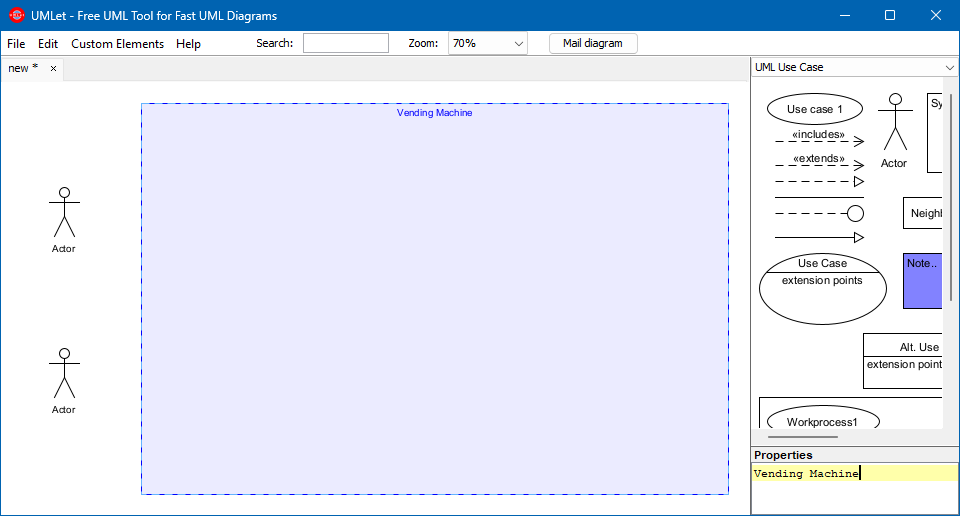
1. Pertama-tama, *drag* dan *drop* entitas sistem menggunakan *template* **SimpleClass**.



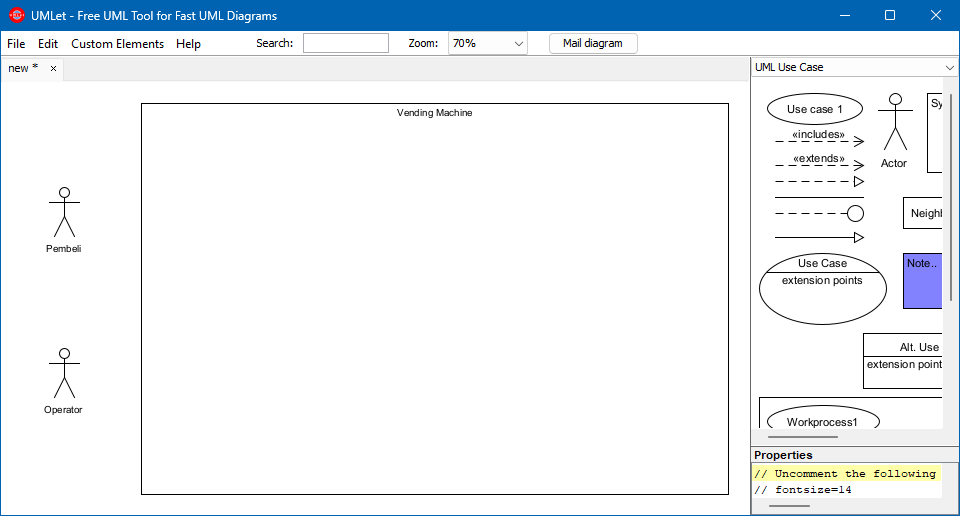
1. Tambahkan dua buah aktor dengan *template* **Actor**.



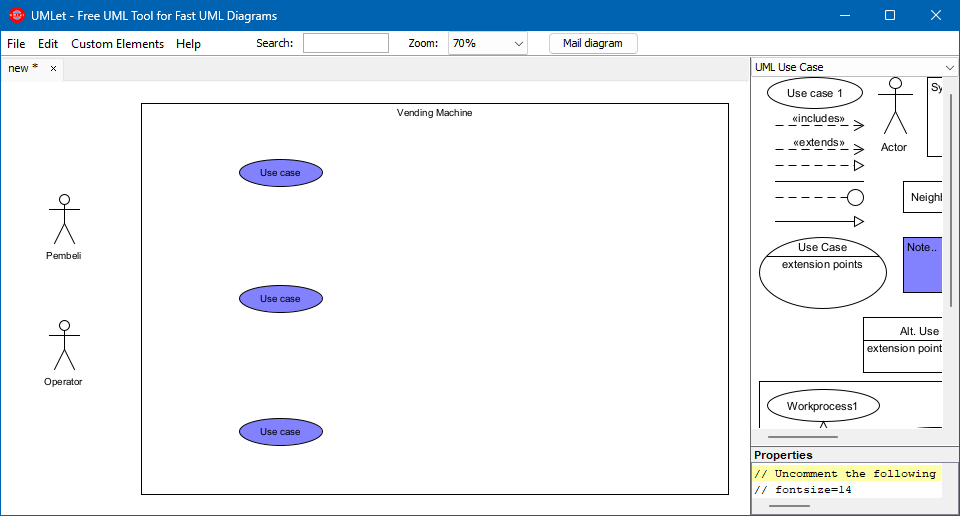
1. Ubah nama sistem menjadi **Vending Machine** sesuai dengan topik kasusnya.

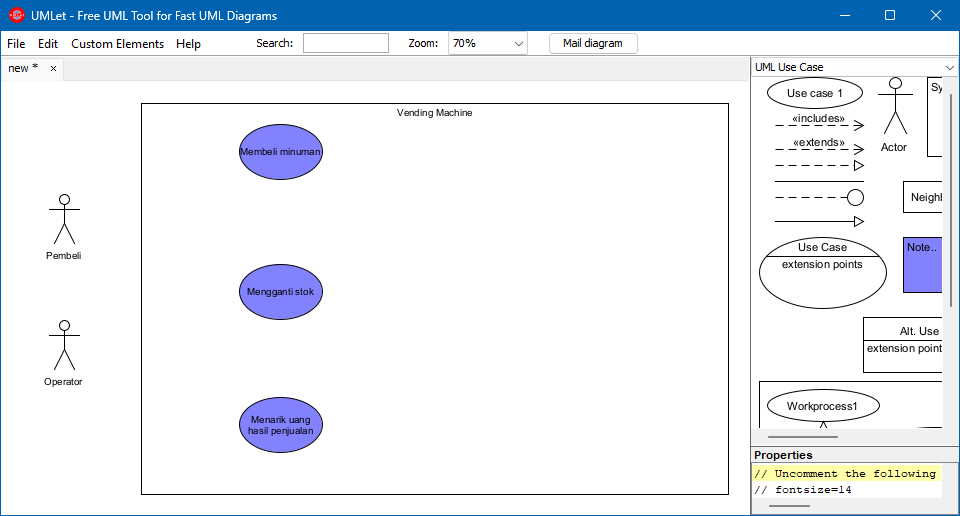


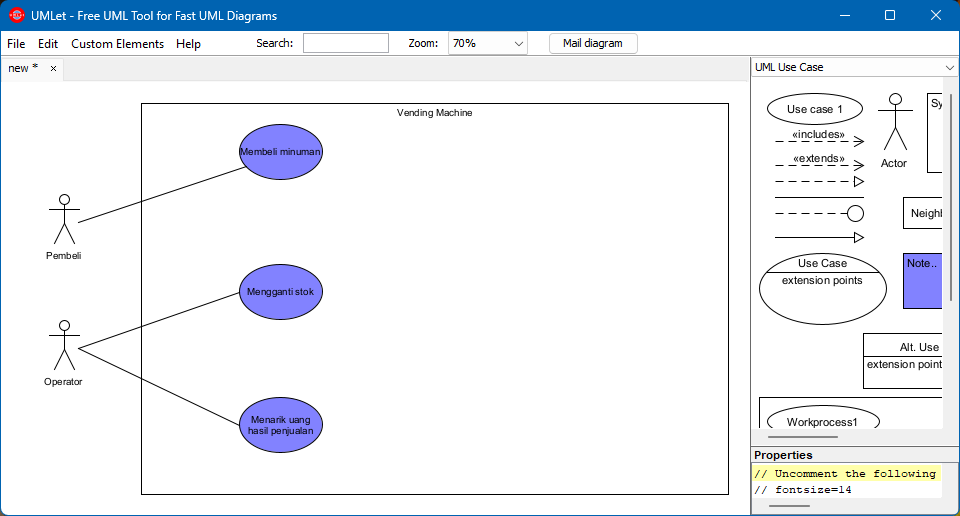
1. Ubah nama Actor menjadi **Pembeli** dan **Operator**.



1. *Use case* langsung dapat diidentifikasikan dengan menurunkan tema interaksi utama antara aktor dan sistem. Buat tiga *use case* baru menggunakan template **Use Case**. *Case* tersebut misalkan Untuk apa pembeli berinteraksi dengan *vending machine*? Untuk **membeli minuman**. Untuk apa operator berinteraksi dengan *vending machine*? Untuk **mengganti stok** dan/atau **menarik uang hasil penjualan**. Kemudian hubungkan aktor dengan *case*-nya.







1. Kemudian identifikasi *use case* bawaan (<<include>>) dengan *break down* terhadap *use case* utama.
2. Apa saja skenario dalam kegiatan “Membeli Minuman”?

* Memasukkan uang
* Memilih minuman
* Mendapatkan minuman

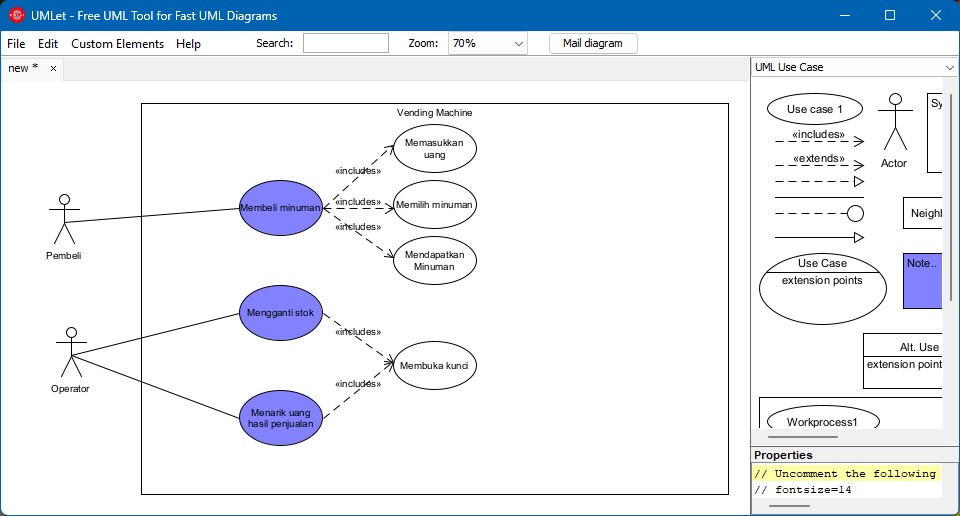
1. Apa saja skenario dalam kegiatan “Mengganti Stok”?

* Membuka kunci
* Mengganti stok(=skenario utama)

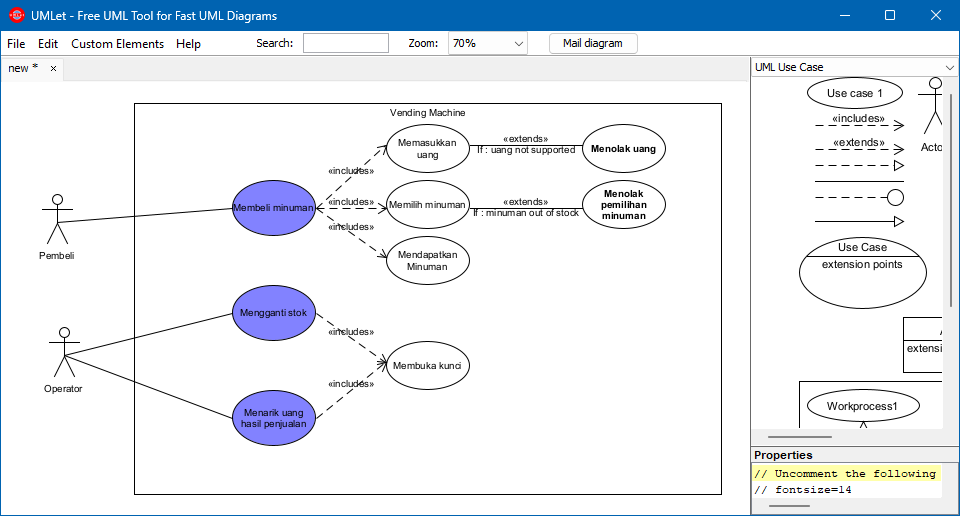
1. Apa saja skenario dalam kegiatan “Menarik Uang Hasil Penjualan”?

* Membuka kunci
* Menarik uang penjualan(=skenario utama)

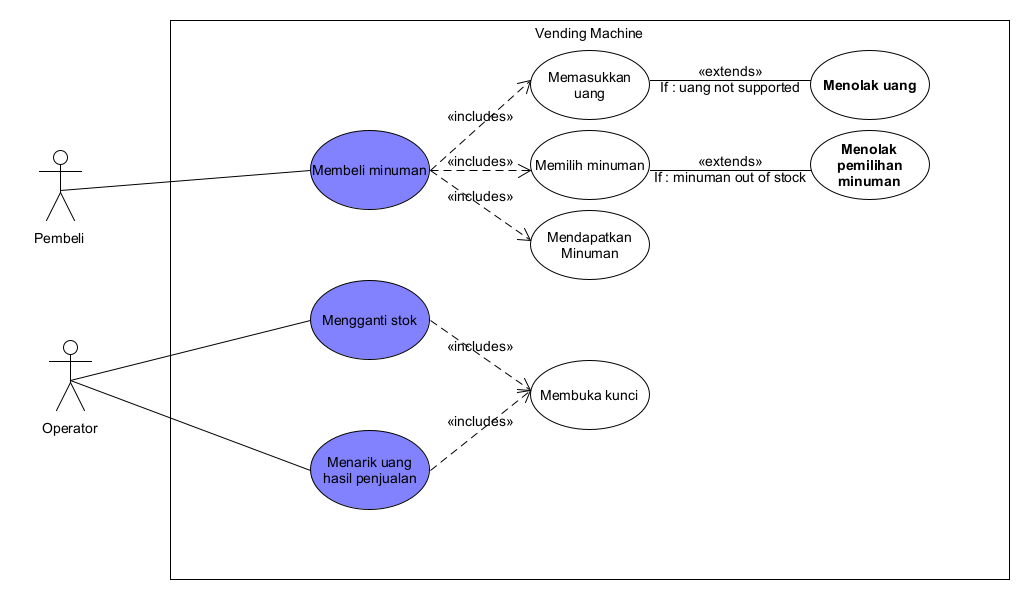
1. Gambar ke dalam *use case* dengan template **Use Case** dan hubungkan dengan panah **<<include>>**.



1. Dalam kondisi khusus, *use case* spesifik harus memenuhi sebuah skenario lain untuk dapat berjalan. Kasus spesifik di kasus ini antara lain :
2. **Menolak uang** jika pada skenario “Memasukkan uang” pembeli memasukkan jenis uang yang tidak dapat terdeteksi sistem.
3. **Menolak pemilihan** minuman jika pada skenario “Memilih minuman” pembeli memilih minuman yang *out* *of stock*.
4. Gambar ke dalam *use case* dengan template **Use Case**, hubungkan dengan panah **<<extend>>**, dan tambahkan skenario spesifik.



**Hasil Percobaan :**



1. **Analisis Percobaan Vending Machine**

Praktikan mencoba mendeskripsikan atau maping berupa *use case* dari kasus *vending machine*. Mula-mula diidentifikasi aktor pemeran langsung dalam ekosistem atau lingkungan aktivitas di *vending machine*. Kemudian dengan adanya *field* **SimpleCase**, letakkan beberapa *use* *case* dengan *breakdown*-nya.

Kasus *vending machine* jika dilihat dari diagram berisi tentang seorang pembeli dan operator sama-sama berinteraksi dengan sistem. Sistem tersebut katakanlah *vending machine*. Sebagai **pembeli**, hal yang mungkin dilakukan adalah **membeli minuman**. Dari kegiatan **membeli minuman** itu sendiri dibuat turunan kegiatannya yang mungkin. Ketika pembeli datang, hal yang dilakukan adalah **memasukkan uang**, kemudian **memilih minuman**, dan akhirnya **mendapatkan minuman** tersebut. Kasus ini bisa dikatakan *include relationship*, karena kegiatan kegiatan aktivitas atau kasus ini tidak dapat terjadi jika tidak ada kasus seseorang **membeli minuman**. Artinya, pembeli jika mau **membeli minuman** setidaknya melalui aktivitas tersebut. Dari uraian di atas, masih bisa di-*breakdown* kembali. Misal, ketika pembeli **membeli minuman** kemudian **memasukkan uang**, *worst case* yang bisa terjadi adalah uang tersebut ditolak mesin, atau mesin **menolak uang**. Dan jika **memilih minuman** bisa saja ditolak karena habis atau lain suatu hal, sehingga sistem mengeluarkan *statement* atau tindakan **menolak pemilihan minuman**. Kedua terakhir adalah *case* spesifik dan tergolong *extend relationship*, karena *case* ini tergolong bahwa *case* yang terhubung sebelumnya dapat berdiri sendiri, alias tidak harus melalui *case* ini.

Sebagai **operator**, hanya memungkinkan dapat di *breakdown* menjadi **mengganti stok** dan **menarik uang hasil penjualan**. Dari *case* ini keduanya harus **membuka kunci** *vending machine* dulu. Uraian di atas sedikit alur dari penjelasan percobaan yang dilakukan.

1. **Percobaan *Use Case* Diagram untuk Mesin ATM**

Sebuah mesin ATM beroperasi 24 jam untuk melayani nasabah yang akan melakukan transaksi perbankan. Transaksi yang dapat dilayani oleh ATM tersebut hanyalah penarikan tunai dan informasi saldo. Seperti layaknya mesin ATM lainnya, mesin tersebut akan melakukan verifikasi keamanan menggunakan kode PIN. Jika pengguna salah memasukkan PIN sebanyak 3 kali, maka kartu ATM yang digunakan akan “ditelan” oleh mesin tersebut. Uang tunai dalam mesin ATM tersebut diisi ulang rutin 2 hari sekali, maupun insidental jika stok uang tunai habis sebelum waktunya.

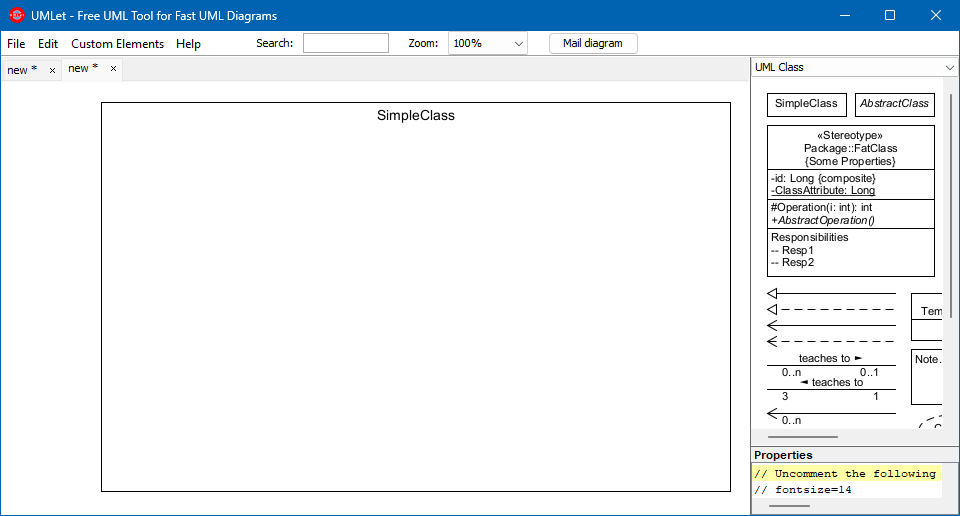
Lakukan analisis terhadap aktor & skenario yang ada, dan buatlah use case diagram dari kasus tersebut!

**Jawaban**

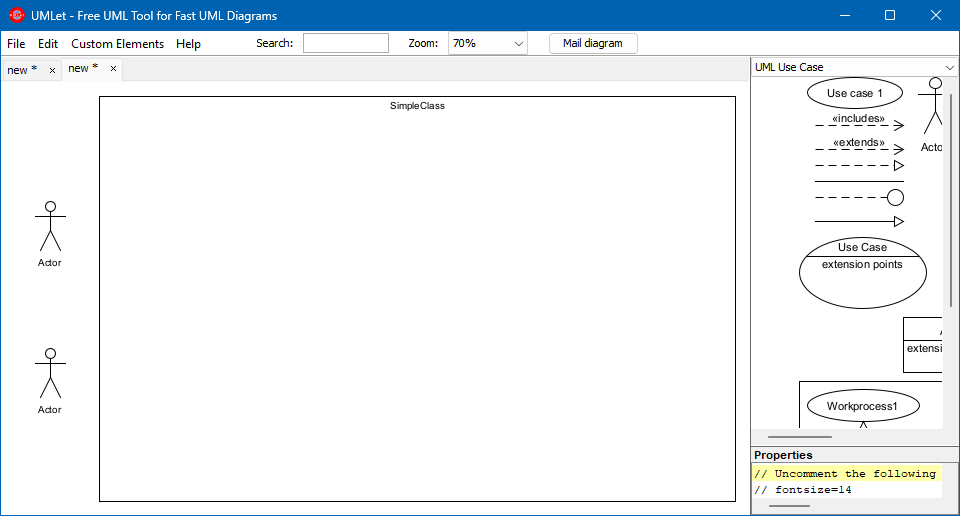
Identifikasikan aktor dengan kata ganti orang atau entitas eksternal sistem. Dalam kasus ini, digunakan kata ganti **Nasabah** dan **Petugas Bank**.

**Langkah-langkah** :

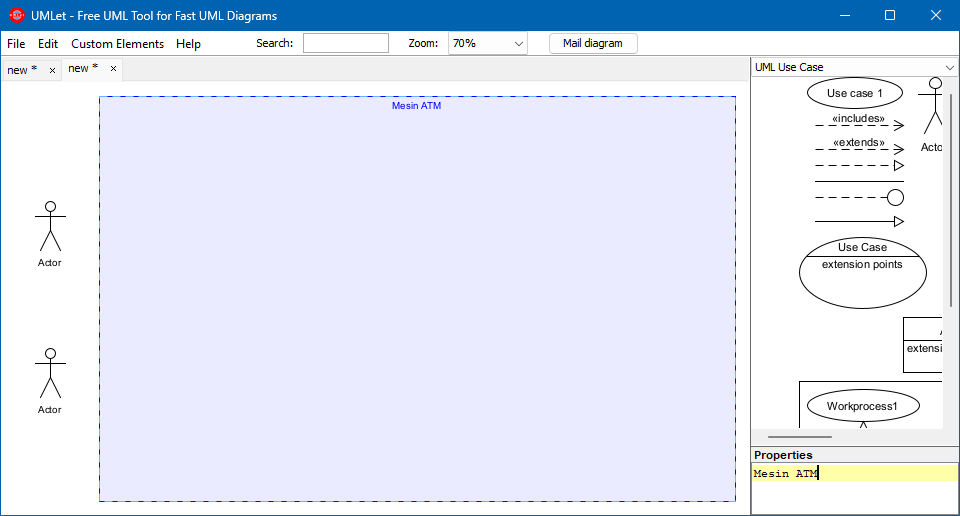
1. Pertama-tama, *drag* dan *drop* entitas sistem menggunakan *template* **SimpleClass**.



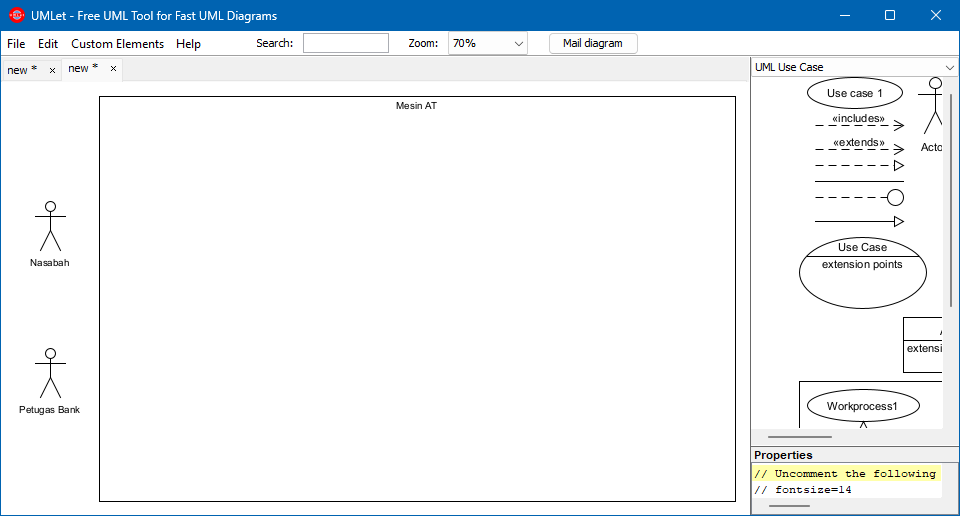
1. Tambahkan dua buah aktor dengan *template* **Actor**.



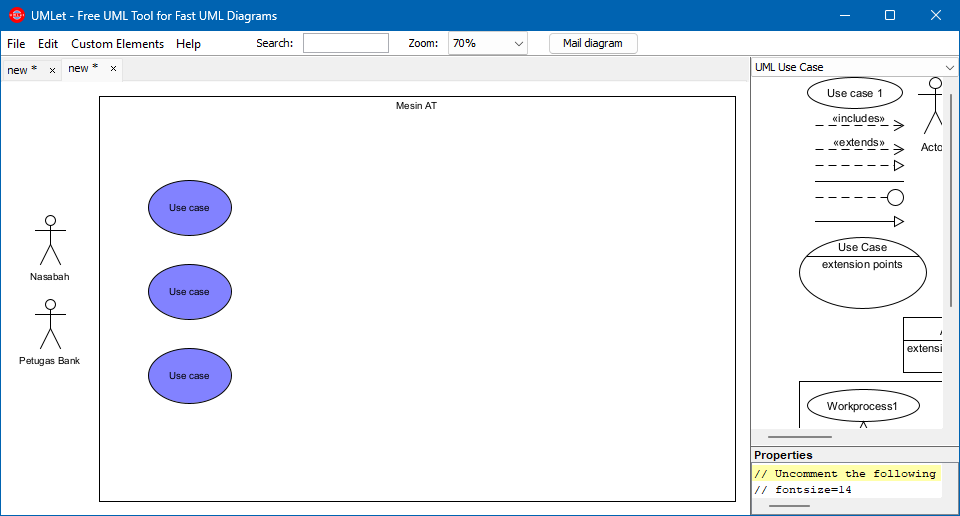
1. Ubah nama sistem menjadi **Mesin ATM** sesuai dengan topik kasusnya.

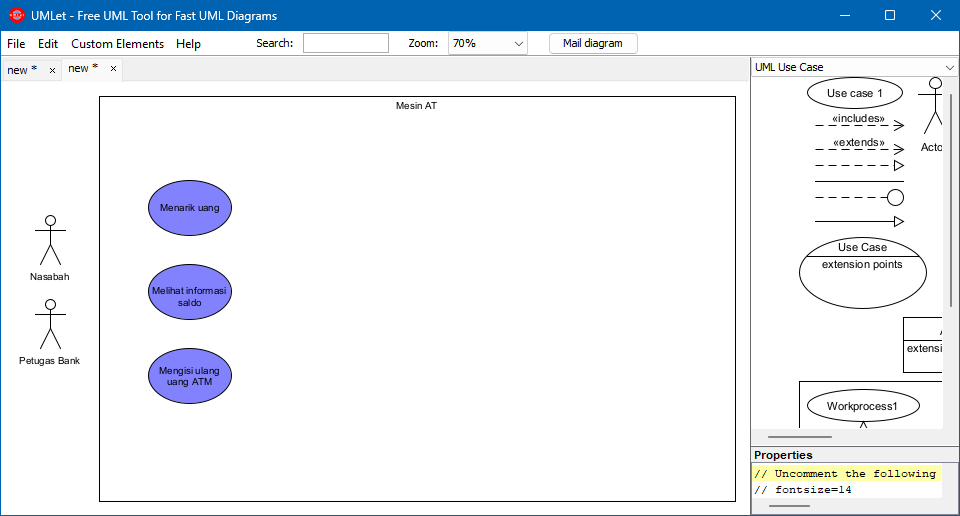


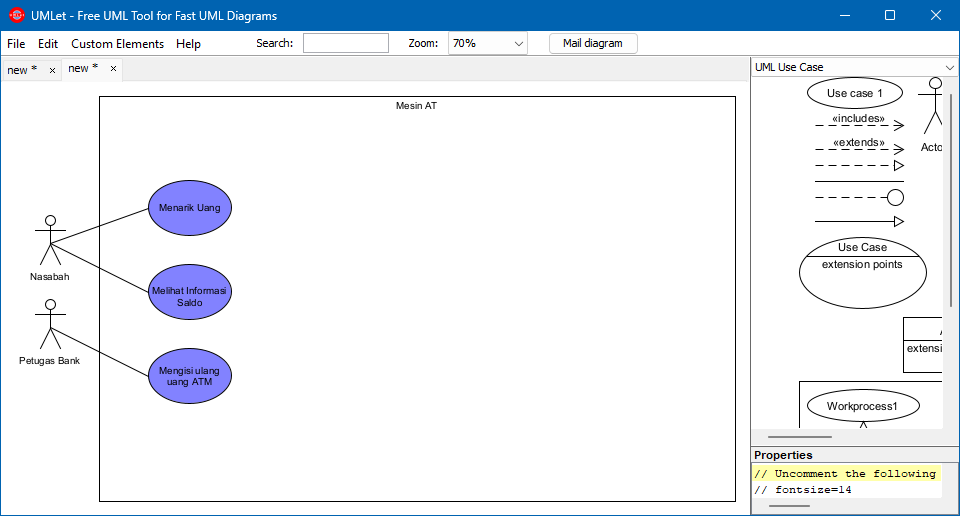
1. Ubah nama Actor menjadi **Nasabah** dan **Petugas Bank**.



1. *Use case* langsung dapat diidentifikasikan dengan menurunkan tema interaksi utama antara aktor dan sistem. Buat tiga *use case* baru menggunakan template **Use Case**. *Case* tersebut misalkan Untuk apa nasabah berinteraksi dengan mesin ATM? Untuk **menarik uang** dan **melihat informasi saldo**. Untuk apa petugas bank berinteraksi dengan mesin ATM? Untuk **mengisi ulang uang ATM**. Kemudian hubungkan aktor dengan *case*-nya.







1. Kemudian identifikasi *use case* bawaan (<<include>>) dengan *break down* terhadap *use case* utama.
2. Apa saja skenario dalam kegiatan “Menarik Uang”?

* Memasukkan kartu
* Masukkan PIN
* Memasukkan nominal
* Menarik uang(=skenario utama)

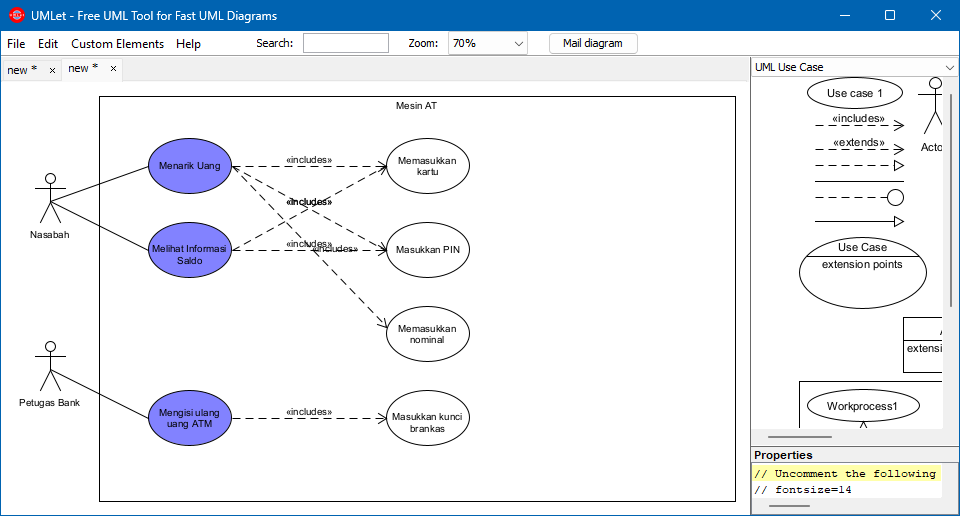
1. Apa saja skenario dalam kegiatan “Melihat Informasi Saldo”?

* Memasukkan kartu
* Masukkan PIN
* Melihat informasi saldo(=skenario utama)

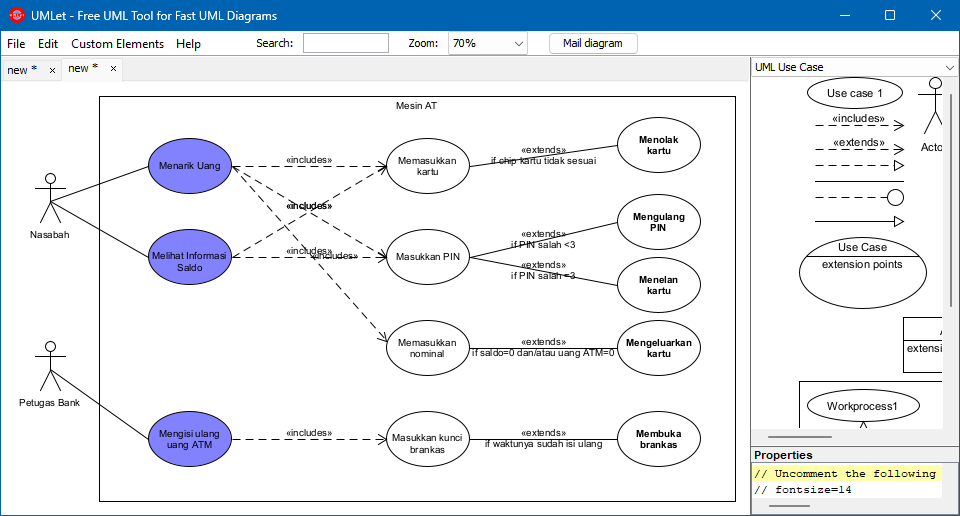
1. Apa saja skenario dalam kegiatan “Mengisi ulang uang ATM”?

* Masukkan kunci brankas
* Mengisi ulang uang ATM(=skenario utama)

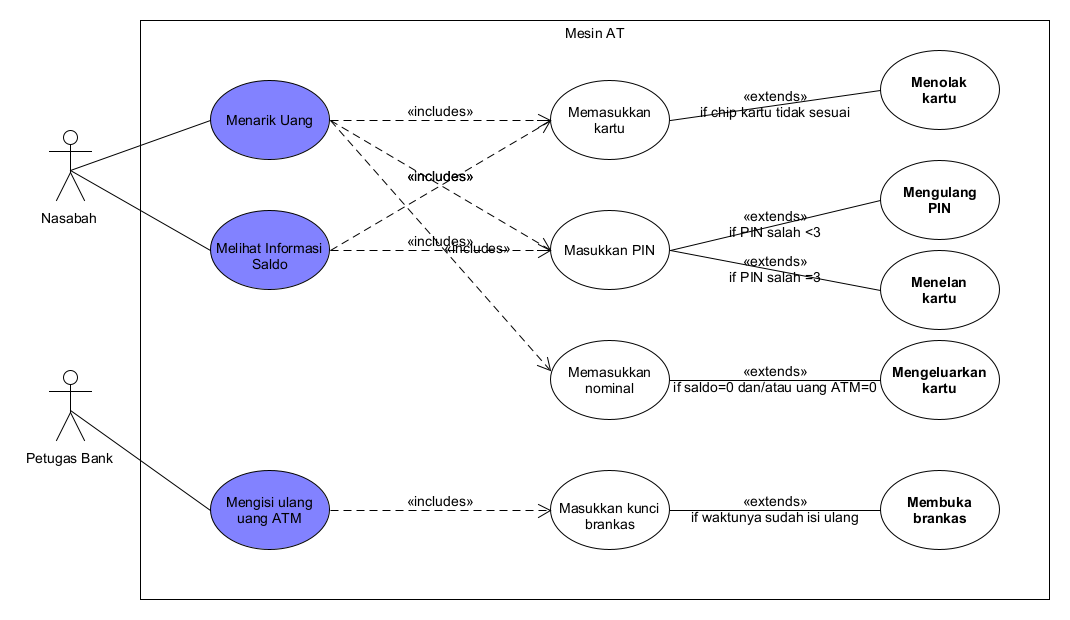
1. Gambar ke dalam *use case* dengan template **Use Case** dan hubungkan dengan panah **<<include>>**.



1. Dalam kondisi khusus, *use case* spesifik harus memenuhi sebuah skenario lain untuk dapat berjalan. Kasus spesifik di kasus ini antara lain :
2. **Menolak kartu** jika pada skenario “Memasukkan kartu” nasabah memasukkan kartu dengan chip yang tidak sesuai.
3. **Mengulang PIN** jika pada skenario “Memasukkan PIN” nasabah salah memasukkan PIN kurang dari tiga kali.
4. **Menelan kartu** jika pada skenario “Memasukkan PIN” nasabah salah memasukkan PIN sebanyak tiga kali.
5. **Mengeluarkan kartu** jika pada skenario “Memasukkan nominal” saldo tidak mencukupi dan/atau uang di ATM tidak mencukupi.
6. **Membuka brankas** jika pada skenario “Masukkan kunci brankas” petugas bank sudah waktunya mengisi ulang.
7. Gambar ke dalam *use case* dengan template **Use Case**, hubungkan dengan panah **<<extend>>**, dan tambahkan skenario spesifik.



**Hasil Percobaan :**

****

1. **Analisis Percobaan Mesin ATM**

Praktikan mencoba mendeskripsikan atau maping berupa *use case* dari kasusmesin ATM. Mula-mula diidentifikasi aktor pemeran langsung dalam ekosistem atau lingkungan aktivitas di mesin ATM. Kemudian dengan adanya *field* **SimpleCase**, letakkan beberapa *use* *case* dengan *breakdown*-nya.

Kasus mesin ATM jika dilihat dari diagram berisi tentang seorang nasabah dan petugas bank sama-sama berinteraksi dengan sistem. Sistem tersebut katakanlah mesin ATM. Sebagai **nasabah**, hal yang mungkin dilakukan adalah **menarik uang** dan  **melihat informasi saldo**. Dari kegiatan **menarik uang** itu sendiri dibuat turunan kegiatannya yang mungkin. Ketika nasabah datang, hal yang dilakukan adalah **memasukkan kartu**, kemudian **memasukkan PIN**, dan akhirnya **memasukkan nominal** tersebut. Kasus ini bisa dikatakan *include relationship*, karena kegiatan kegiatan aktivitas atau kasus ini tidak dapat terjadi jika tidak ada kasus seseorang **menarik uang**. Artinya, nasabah jika mau **menarik uang** setidaknya melalui aktivitas tersebut. Dari uraian di atas, masih bisa di-*breakdown* kembali. Misal, ketika pembeli **menarik uang** kemudian **memasukkan kartu**, *worst case* yang bisa terjadi adalah kartu ditolak, atau mesin **menolak kartu**. Jika dalam proses menarik uang maupun mengecek saldo, kemudian **memasukkan PIN**, sistem bisa **mengulang PIN** dan **menelan kartu** jika salah memasukkan PIN. Dan jika **memasukkan nominal** bisa saja ditolak karena saldo habis atau uang dalam ATM habis, sehingga sistem mengeluarkan tindakan mengeluarkan kartu. Beberapa penjelasan di atas adalah *case* spesifik dan tergolong *extend relationship*, karena *case* ini tergolong bahwa *case* yang terhubung sebelumnya dapat berdiri sendiri, alias tidak harus melalui *case* ini.

Sebagai **petugas bank**, hanya memungkinkan dapat di *breakdown* menjadi **mengisi ulang uang ATM** dan **memasukkan kunci brankas**. Setelah itu, ketika memasukkan kunci brankas, brankas dibuka ketika sudah waktunya.

1. **Kesimpulan**

Dalam suatu *case*, lihat dan perhatikan pelaku dalam kasus tersebut. Pilah dan pilih pelaku tersebut berkontribusi dalam hal apa dan dapat mengakibatkan apa dalam sistem. Hal ini dapat diturunkan kembali menjadi *case* yang bersifat *include* maupun *extend*. Dari pilahan *include* didapatkan *case* yang membuat *case* sebelumnya bergantungan. Kemudian jika *extend*, dia berdiri sendiri.

1. **Percobaan *Use Case* Diagram untuk Tugas Besar**

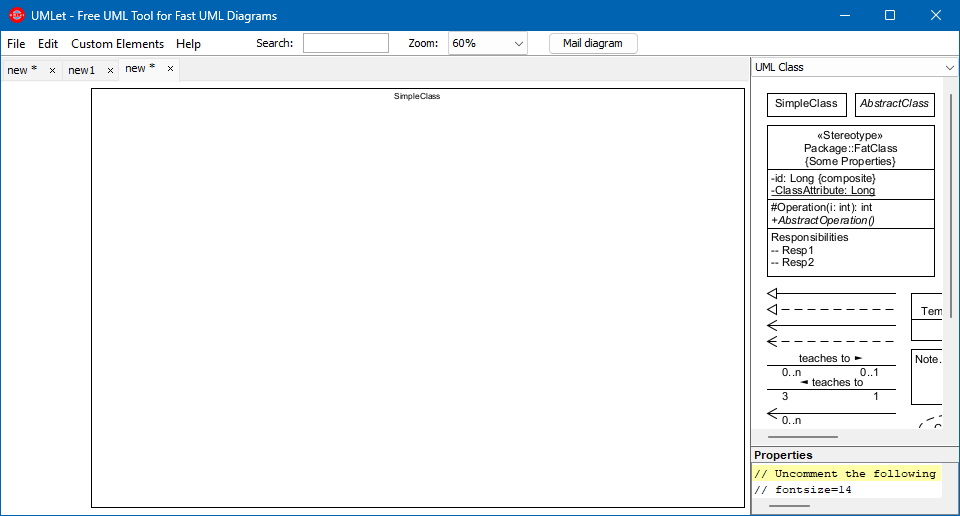
Sebuah cafe di Surakarta bernama DNA SPOT Cafe. Cafe tersebut membutuhkan suatu sistem booking secara online via WhatsApp dan Website. Pembeli dapat memesan melalui WhatsApp dan membayar biaya *down payment*. Admin akan menginputkan baik pesanan makan maupun minuman. Admin dapat membalas pesan WhatsApp. Admin dapat mengetahui ruangan yang belum dipesan.

**Jawaban**

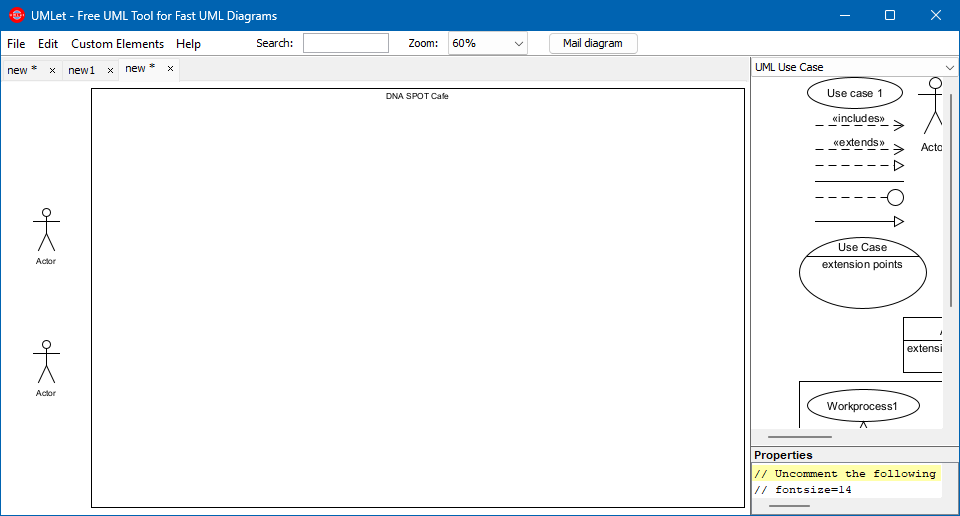
Identifikasikan aktor dengan kata ganti orang atau entitas eksternal sistem. Dalam kasus ini, digunakan kata ganti **Pembeli**, **Admin**, dan **Developer**.

**Langkah-langkah** :

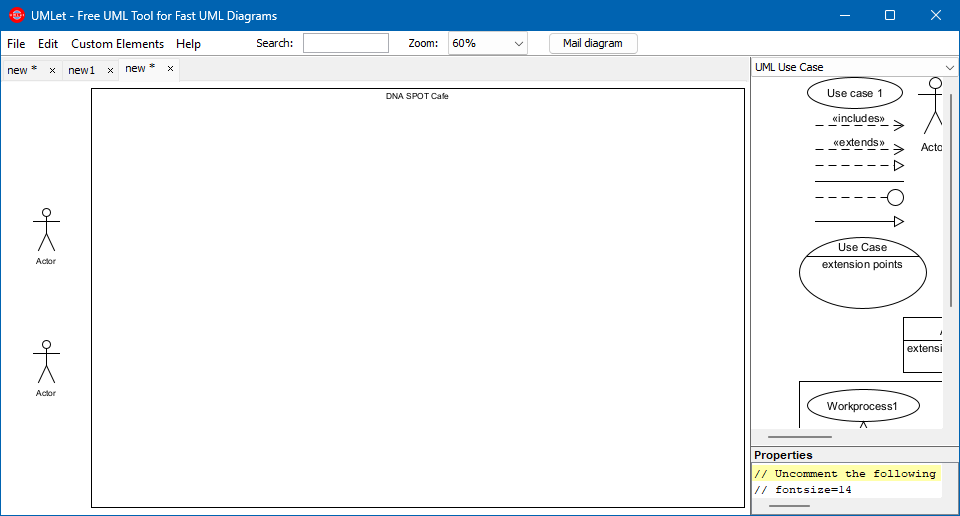
1. Pertama-tama, *drag* dan *drop* entitas sistem menggunakan *template* **SimpleClass**.



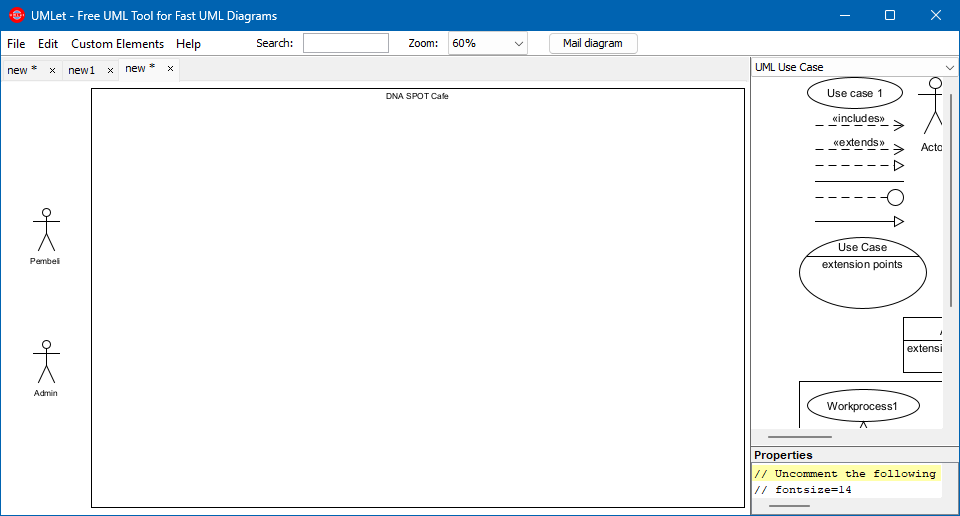
1. Tambahkan du buah aktor dengan *template* **Actor**.



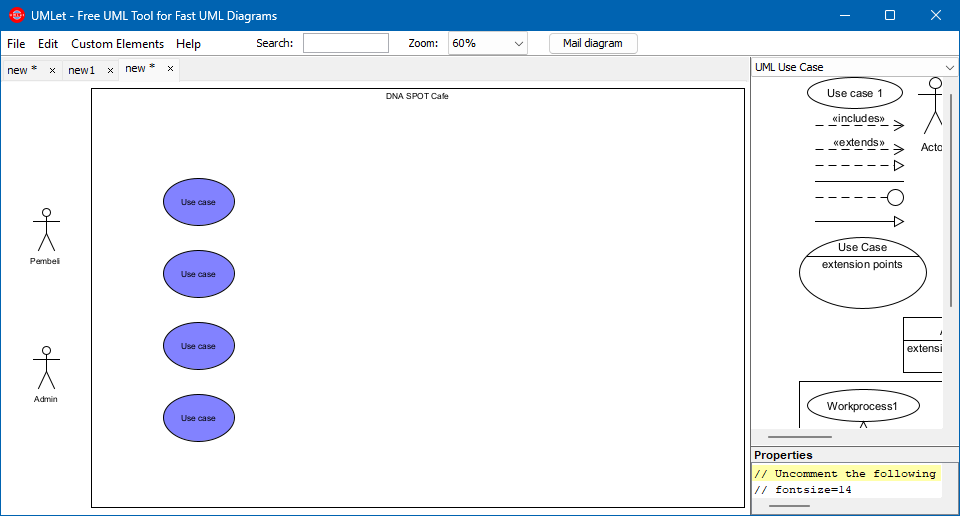
1. Ubah nama sistem menjadi **DNA SPOT Cafe** sesuai dengan topik kasusnya.

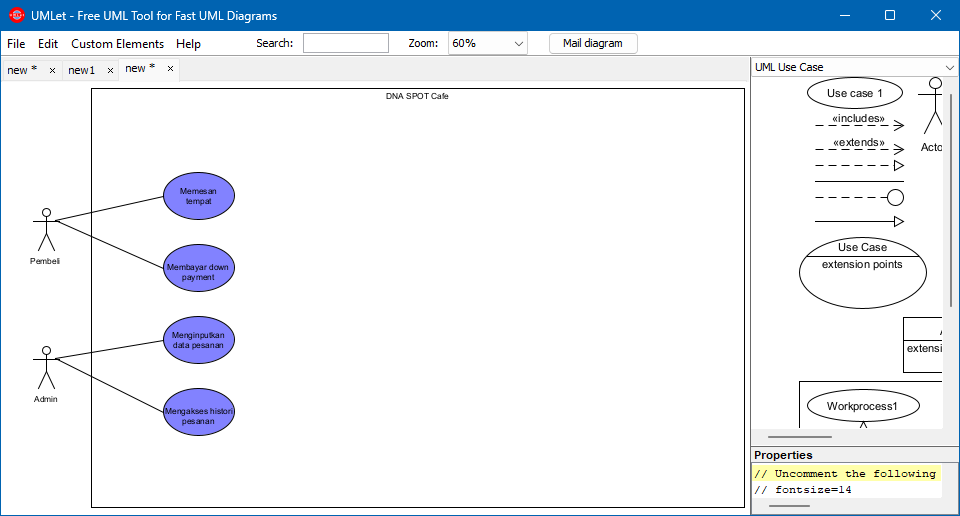


1. Ubah nama Actor menjadi **Pembeli** dan **Admin**.



1. *Use case* langsung dapat diidentifikasikan dengan menurunkan tema interaksi utama antara aktor dan sistem. Buat tiga *use case* baru menggunakan template **Use Case**. *Case* tersebut misalkan Untuk apa pembeli berinteraksi dengan sistem? Untuk **memesan tempat** dan **membayar down payment**. Untuk apa admin berinteraksi dengan sistem? Untuk **menginputkan data pesanan** dan **mengakses histori pesanan**. Kemudian hubungkan aktor dengan *case*-nya.





1. Kemudian identifikasi *use case* bawaan (<<include>>) dengan *break down* terhadap *use case* utama.
2. Apa saja skenario dalam kegiatan “Memesan Tempat”?

* Masuk ke pesan singkat WhatsApp
* Memilih tempat dan makanan minuman
* Memesan tempat(=skenario utama)

1. Apa saja skenario dalam kegiatan “Membayar Down Payment”?

* Masuk ke mobile banking
* Memasukkan nomor rekening management
* Membayar Down Payment(=skenario utama)

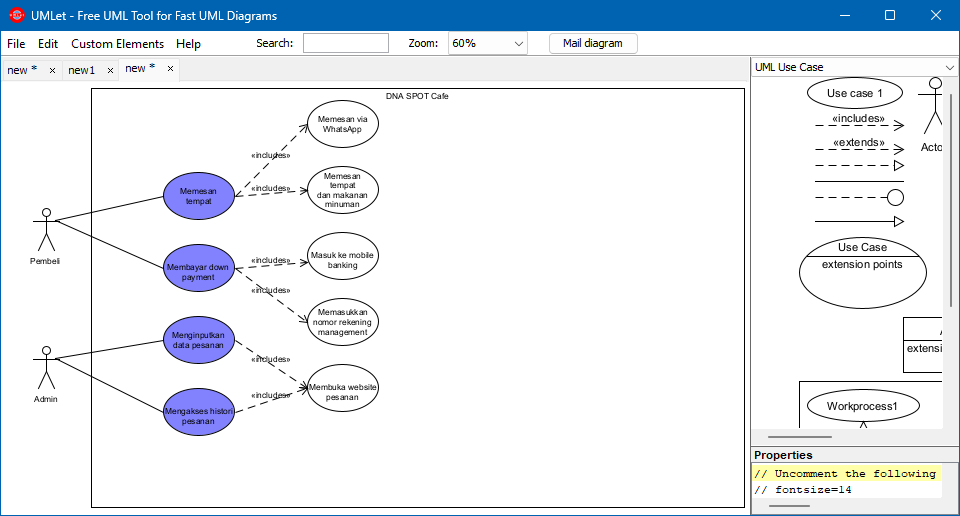
1. Apa saja skenario dalam kegiatan “Menginputkan Data Pesanan”?

* Membuka website pesanan
* Menginputkan data pesanan(=skenario utama)

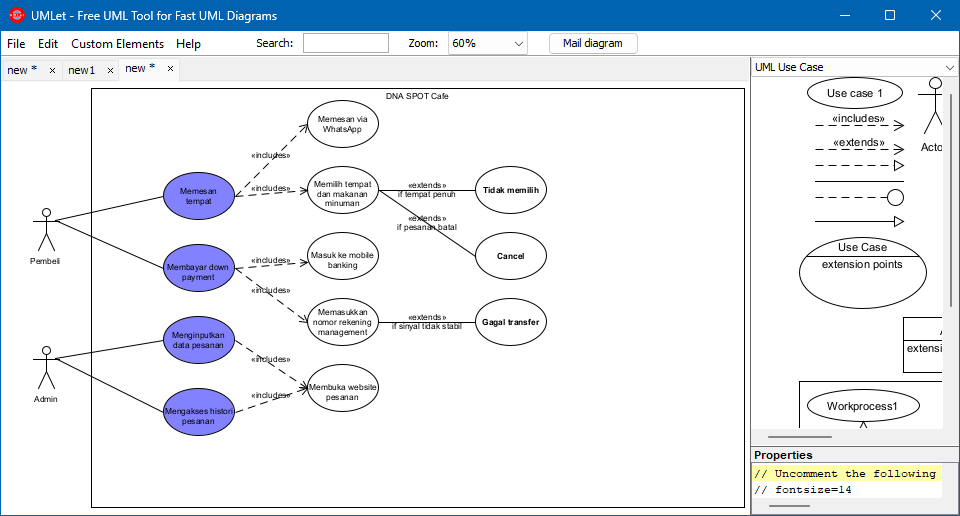
1. Apa saja skenario dalam kegiatan “Mengakses Histori Pesanan”?

* Membuka website pesanan
* Mengakses histori pesanan(=skenario utama)

1. Gambar ke dalam *use case* dengan template **Use Case** dan hubungkan dengan panah **<<include>>**.



1. Dalam kondisi khusus, *use case* spesifik harus memenuhi sebuah skenario lain untuk dapat berjalan. Kasus spesifik di kasus ini antara lain :
2. **Tidak memilih** jika pada skenario “Memilih tempat dan makanan minuman” cafe kehabisan tempat.
3. **Cancel** jika pada skenario “Memilih tempat dan makanan minuman” pembeli membatalkan pesanan.
4. **Gagal Transfer** jika pada skenario “Memasukkan nomor rekening management” sinyal pembeli kurang bagus.
5. Gambar ke dalam *use case* dengan template **Use Case**, hubungkan dengan panah **<<extend>>**, dan tambahkan skenario spesifik.



**Hasil Percobaan :**

