**Nama : Nurul Hana Sintya**

**Nim : 124180036**

**Kelas : UKPL-B**

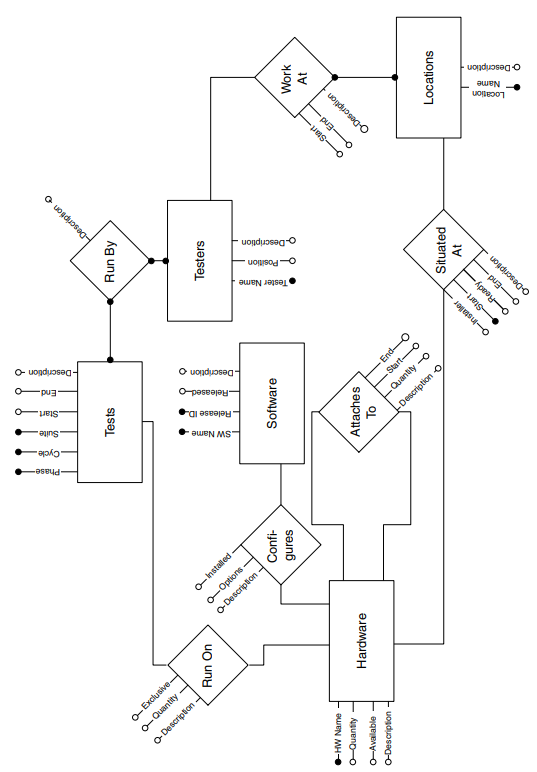
**TUGAS BAB 9**

1. **Jelaskan Gambar Hubungan antara Test tracking dan Laporan seperti gambar diatas !**

**Jawab:**

1. **Jelaskan ERD pada Penerapan Basis Data Logistik diatas !**

**Jawab:**



Gambar diatas menunjukkan diagram entity-relationship (E-R) untuk database.

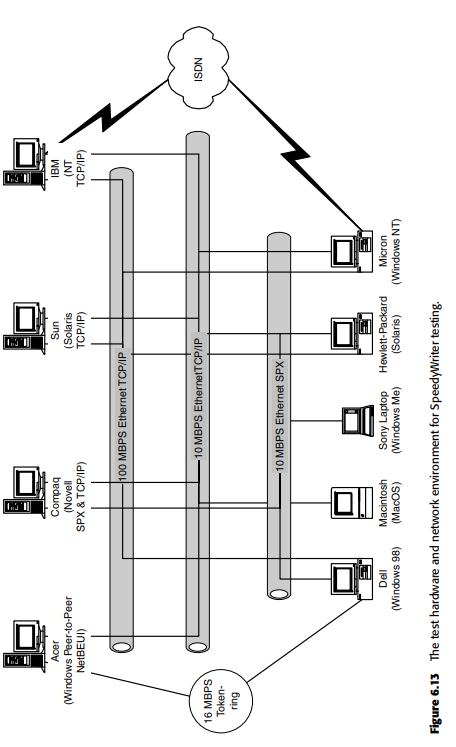
* ***The rectangles*** mewakili entitas, yang merupakan objek dari beberapa jenis. Setiap entitas memiliki satu set properti, diwakili oleh garis berlabel. Satu (atau lebih) dari properti ini, yang disebut kunci, secara unik mengidentifikasi entitas individu.
* ***Lines with solid endpoints*** menunjukkan properti kunci, sedangkan garis dengan titik akhir terbuka menunjukkan properti bukan kunci. Misalnya, Nama adalah properti kunci dari entitas Perangkat Keras, sedangkan Kuantitas adalah properti bukan kunci dari entitas ini.
* ***The diamonds*** mewakili hubungan antara pasangan entitas. Hubungan juga dapat memiliki sifat. (Namun, tidak semua hubungan memiliki properti; hubungan diidentifikasi oleh entitas yang mereka kaitkan.) Ada tiga jenis hubungan yaitu one-to-one, one-to-many, and many-to-many.
* ***The Tests entity*** memiliki dengan enam properti, muncul di bagian tengah atas bagan.

1. Properti Phase dapat berupa Component, Integration, atau System.
2. Cycle adalah nomor urut dari test cycle pada fase tersebut; Saya menggunakan istilah cycle daripada passes karena kami akan melacak revisi perangkat lunak dengan alat ini, yang merupakan dasar dari test cycle dalam pendekatan saya.
3. Properti Suite adalah nama suite. (Atau, Anda dapat menggunakan nomor ID untuk properti Suite, meskipun angkanya kurang deskriptif.) Tiga properti pertama yang digabungkan secara unik mengidentifikasi pengujian individual, seperti pada Tes Integrasi GUI/Edit Engine.
4. Properti start dan end menunjukkan tanggal pelaksanaan terjadwal untuk suite tersebut dalam siklus tersebut.
5. Description mencakup informasi terkait lainnya; ada bidang deskripsi yang disertakan untuk catatan bentuk bebas dengan setiap entitas dan hubungan.

* ***Entities Tests and Testers*** ditautkan oleh hubungan **Run By** dengan kata lain, pengujian dijalankan oleh penguji. Penguji secara unik diidentifikasi dengan nama mereka. Penguji memiliki posisi, umumnya baik insinyur pengujian atau teknisi pengujian. Karena setiap pengujian dijalankan oleh satu atau lebih penguji, dan setiap penguji menjalankan satu atau lebih pengujian, hubungannya adalah many-to-many.
* ***Testers work at locations*** yaitu entitas testers dan location ditautkan oleh hubungan **Work At**. Dalam beberapa kasus, lokasi bersifat sementara, dalam kerangka waktu proyek, sehingga hubungannya mencakup tanggal mulai dan tanggal berakhir. Penguji mungkin bekerja di lebih dari satu lokasi selama periode waktu tertentu.
* Tests **run on** one or more items of hardware, yang masing-masing menjalankan satu atau lebih pengujian. Pengujian memerlukan jumlah tertentu dari setiap item perangkat keras, biasanya satu tetapi terkadang lebih, dan hubungan pengujian dengan perangkat keras dapat bersifat eksklusif atau non-eksklusif, tergantung pada apakah pengujian memerlukan penggunaan khusus perangkat keras tersebut selama pelaksanaannya.
* Software **configures** most hardware items yaitu, kombinasi tertentu dari satu atau lebih item perangkat lunak yang diinstal pada perangkat keras, dan cara pemasangannya, menentukan fungsionalitas perangkat keras. Perangkat lunak yang sama mungkin juga diinstal pada beberapa item perangkat keras; misalnya, Anda mungkin menjalankan Linux di beberapa server di peternakan server Anda. Entitas Perangkat Lunak memiliki empat properti yang menarik: Name, Release ID (the release number), Released (the release date), and Description.
* In a networked or telecommunications environment, a given piece of hardware **attaches** to one or more other pieces of hardware. Router, hub, dan switch terhubung ke banyak perangkat sekaligus; modem, keyboard, dan mouse terpasang ke satu komputer pada satu waktu. Lampiran tersebut memiliki tanggal mulai dan berakhir, dan jumlah yang diperlukan juga relevan.
* Finally, hardware items are situated at a particular location during certain periods of time. In the **Situated At** relationship, an installer sets up the hardware, starting on a particular date. Perangkat keras akan siap untuk diuji pada tanggal tidak lama kemudian, dan perangkat keras akan mengakhiri kunjungannya di lokasi tersebut di kemudian hari. Lebih dari satu perangkat keras mungkin terletak di lokasi tertentu, dan perangkat keras mungkin berpindah-pindah selama upaya pengujian.

1. **Jelaskan contoh pengujian test hardware dan jaringan diatas !**

**Jawab:**



Mari kita asumsikan bahwa Anda ingin mencakup empat jenis jaringan logis: Microsoft peer-to peer (NetBEUI), Novell (SPX dan TCP/IP), Solaris (NFS dengan TCP/IP), dan Windows NT (TCP/IP). Misalkan semuanya kecuali server peer-to-peer Microsoft sudah aktif dan berjalan di gedung; Anda kemudian dapat menganggarkan sistem Acer yang murah untuk menjadi tuan rumah yang bekerja sama. Sistem tersebut, dengan cara yang berbeda, menghubungkan ke lima jenis jaringan fisik yang berbeda: 100 Mbps Ethernet TCP/IP, 10 Mbps Ethernet TCP/IP, 10 Mbps Ethernet SPX, 16 Mbps token-ring NetBEUI, dan jaringan dial-up ISDN TCP/IP. (Penting untuk menyertakan tipe jaringan fisik yang berbeda karena koneksi jaringan yang berbeda memerlukan driver yang berbeda di sisi klien, yang dapat bersinggungan dengan operasi aplikasi meskipun mereka tidak seharusnya melakukannya.) Gamar diatas menunjukkan konfigurasi lab uji dengan klien ini , server, dan jenis jaringan.

Meskipun Anda sekarang telah mendefinisikan jaringan uji tanpa banyak memperhatikan pengujian khusus yang akan Anda jalankan di dalamnya, ini bukan kasus menempatkan kereta di depan kuda. Definisi jaringan pengujian Anda tumbuh dari sasaran cakupan Anda untuk klien, server, dan jenis jaringan. Setelah memenuhi tujuan ini, Anda sekarang dapat mendistribusikan rangkaian pengujian di berbagai konfigurasi klien, baik secara acak atau berdasarkan mana yang menurut Anda paling mungkin untuk menemukan bug terhadap konfigurasi tertentu. Perhatikan bahwa dengan menyebarkan pengujian ke klien, Anda juga dapat mencakup server dan jaringan dengan mudah, asalkan Anda memastikan bahwa pengujian benar-benar menjalankan semua koneksi jaringan yang tersedia untuk klien.

1. **Jelaskan tentang BasisData Pengelola Perubahan Sederhana !**

**Jawab:**

Manajemen perubahan basis data bukanlah tugas yang mudah. Lebih sulit lagi ketika kita berada di awal proyek di mana model data terus berubah. Lebih banyak orang dalam tim adalah kesulitan lain karena kami harus mengoordinasikan pekerjaan kami dan tidak saling mengganggu. Tentu saja, kami dapat mendelegasikan manajemen perubahan basis data ke satu orang, tetapi itu tidak mudah. Selain itu, kita harus menjaga otomatisasi penerapan, karena tanpa otomatisasi kita tidak dapat memiliki integrasi dan pengiriman yang benar-benar berkelanjutan. Bagaimanapun sempurnanya suatu perencanaan, langkah penanganan yang hati-hati, kefektifan kerjasama dengan tim penguji, rekan kerja, dan manajer tetap saja akan menghadapi tanggapan yang tidak diharapkan. Perubahan tidak dapat dihindari dan tidak seluruh perubahan dapat direncanakan sebelumnya. Perubahan seringkali merupakan manifestasi dari proses pembelajaran. Pengelolaan perubahan dapat dilakukan secara formal dengan CCB (change control board) atau secara informal dengan simple change management database. Change Management Database (CMDB) adalah alat yang dapat digunakan organisasi untuk melacak aset TI dan hubungan di antara mereka. Manfaatnya termasuk kontrol logistik, kepatuhan lisensi, dan perhitungan ROI yang akurat.