MODUL MANAJEMEN BASIS DATA DAN APLIKASI DALAM MIK (KODE: MIK351)

Dosen Pengampu:
Noviandi, M.Kom

Prodi D4 Manajemen Informatika Kesehatan
Universitas Esa Unggul
2018

Daftar Isi

Halaman

Modul1

Konsep Database dan Struktur Data

Modul pertama ini terdiri atas dua topic pembahasan, yaitu memahami tentang konsep dasar *database* dan struktur data. Pembahasan pada topic kedua menjelaskan tentang tahap-tahap desain *database* dan penerapannya dengan menggunakan Ms. Access.

Selain itu, secara khusus mahasiswa mampu untuk:

1. Memahami dan menjelaskan konsep *database*
2. Menjelaskan tahap-tahap dalam men-*design* *database* dan menerapkan perancangan *database* dengan menggunakan ms.access.

Topik 1

Konsep Database dan Pemahaman Dasar Struktur Data

Database Secara Umum

Pengertian database (basis data) yang paling sederhana adalah kumpulan dari tabel. Satu tabel mempresentasikan suatu entitas tertentu. Suatu entitas terdiri atas beberapa atribut. Tabel-tabel yang di kumpulkan dalam satu wadah atau container. Wadah inilah yang disebut dengan **database**.

Berikut merupakan beberapa definisi database:

a. Menurut George Tsu-der Chou
   Kumpulan informasi yang diorganisasikan kedalam tatacara khusus.
b. Menurut Anthoni J. Fabbri & A. Robert Schwab
   Sistem berkas yang dirancang terutama untuk meminimalkan pengulangan data
c. Menurut C.J Date
   Sistem terkomputerisasi yang tujuan utamanya adalah memelihara informasi dan membuat informasi itu ada pada saat dibutuhkan
d. Menurut Silberschatz dkk
   Kumpulan data yang saling berelasi dan program yang memungkinkan user untuk mengakses dan memodifikasi data.

Fungsi sistem database adalah

a. Membantu dalam pemeliharaan dan utilitas kumpulan data dalam jumlah yang besar
b. Berfungsi dalam mengelola database, mulai dari meng-*create*sampai dengan proses-proses yang berlaku dalam database tersebut.

Salah satu manfaat database yang paling utama adalah untuk memudahkan dalam mengakses data. Kemudahan pengaksesan data ini adalah sebagai implikasi dari ketentuan data yang merupakan syarat mutlak dari suatu database yang baik. Sebagai ilustrasi akan arti penting database dalam kehidupan sehari-hari adalah kasus lemari pakaian. Apabila kita menaruh pakaian secara sembarangan dan tidak teratur. Misal Anda memasukkan atau meletakkan kaos oblong, sarung, pakaian dalam, kaos kaki, kemeja, baju renang, pakaian tidur, saputangan, celana panjang, celana pendek, selimut, dan yang lainnya dalam satu tempat yang tidak diatur, maka semua akan bercampur baur dan tentunya menjadi tidak teratur.

Konsekuensi dari ketidakteraturan dalam penempatan beberapa jenis barang tentunya akan menyulitkan dalam pencarian atau pemilihan barang yang akan Anda ambil atau Anda pilih. Sebagai contoh Anda akan mencari kaos kaki yang merupakan jenis barang dalam kategori kecil, mungkin Anda harus mengacak acak sarung, selimut, celana, pakaian tidur dan jenis barang lainnya. Hal ini tentunya sangatlah tidak efektif dan efisien.

Beda halnya misalkan Anda menempatkan barang tersebut ke dalam suatu lemari dan mengaturnya sesuai dengan jenis barang tersebut. Misalkan dalam lemari tersebut Anda pilah-pilah menjadi beberapa tempat atau bagian. Anda bisa membagi ke dalam bagian atau tempat untuk:

a. Pakaian dalam (berisi singlet, celana dalam, kaos kaki dan sapu tangan)
b. Kemeja (berisi kemeja lengan panjang, lengan pendek, baju batik dan jas)
c. Celana (berisi celana panjang, celana pendek, training, celana renang)
d. Pakaian tidur (berisi piyama, sarung bantal, seprei dan selimut)
e. Laci (bisa untuk menyimpan buku tabungan, kunci dan barang berharga lainnya)

Contoh diatas bisa di analogikan dengan suatu database ‘Lemari Pakaian’ yang terdiri atas tabel-tabel: ‘Pakaian_Dalam, Kemeja, Celana, Pakaian_Tidur, dan Laci’. Sedangkan jenis barang yang ada merupakan field atau kolom. Jadi dalam Tabel Celana terdiri atas field: ‘celana panjang, celana pendek, training dan celana renang’. Sedangkan banyaknya jumlah celana panjang, celana pendek, training dan celana renang menunjukkan banyaknya record dari jenis barang tersebut.

Semakin teratur dalam menyimpan maka akan semakin mudah dalam melakukan pengakesan. Biasanya semakin teratur penyimpanan semakin banyak usaha yang harus dilakukan. Dengan kata lain keteraturan berbanding lurus dengan kesulitan.

Namun dari keteraturan ini akan memberikan banyak keuntungan, diantaranya adalah kemudahan dalam hal pengaksesan.

Database Secara Umum

Pemodelan data atau model data adalah sekumpulan perangkat konseptual yang menggambarkan data, hubungan antar data, semantik data dan batasan data. Dengan menggunakan model data ini maka akan memberikan kemudahan untuk melakukan analisis dan dapat dilakukan perbaikan sebelum diimplementasikan kedalam DBMS. Pemodelan data terdiri atas beberapa model, yaitu: model hirarkis, jaringan dan relasional serta berorientasi objek.

Model yang sering digunakan adalah model relasional. Model relasional yang terkenal adalah *entity relationship diagram* (ERD). Tiga hal yang paling penting dalam ERD, yaitu entitas, relationship dan atribut.

1. Entitas

   Entitas adalah objek yang ada dan dapat dibedakan dengan objek lain (buku, orang, liburan, absensi). Entitas direpresentasikan dengan menggunakan sekumpulan atribut, entitas orang mempunyai atribut nama, alamat, tanggal lahir, dll.

2. Atribut

   Atribut adalah gambaran dari entitas.

   - Setiap atribut harus dijelaskan dengan suatu nilai, misalnya entitas orang mempunyai atribut nama dengan nilai “noviandi”.
   - Nilai dari atribut tersebut juga dapat diatur. Misalnya, panjang karakter dari nama tidak boleh lebih dari 15 karakter. Pengaturan nilai atribut disebut dengan domain.
   - Nilai atribut dapat bernilai tunggal maupun jamak (multi-valued), sederhana (simple) atau gabungan (composite), kosong (Null), atau harus ada (Not Null), dan Key (Primary atau Foreign Key) atau non key.
   - Dalam atribut ada istilah Stored Attribute, yaitu atribut yang langsung terlihat pada entitas (atribut nama, atribut alamat), derived atribut yaitu

atribut hasil perhitungan dari atribut yang lain (misal atribut umur dihitung dari atribut tanggal lahir).

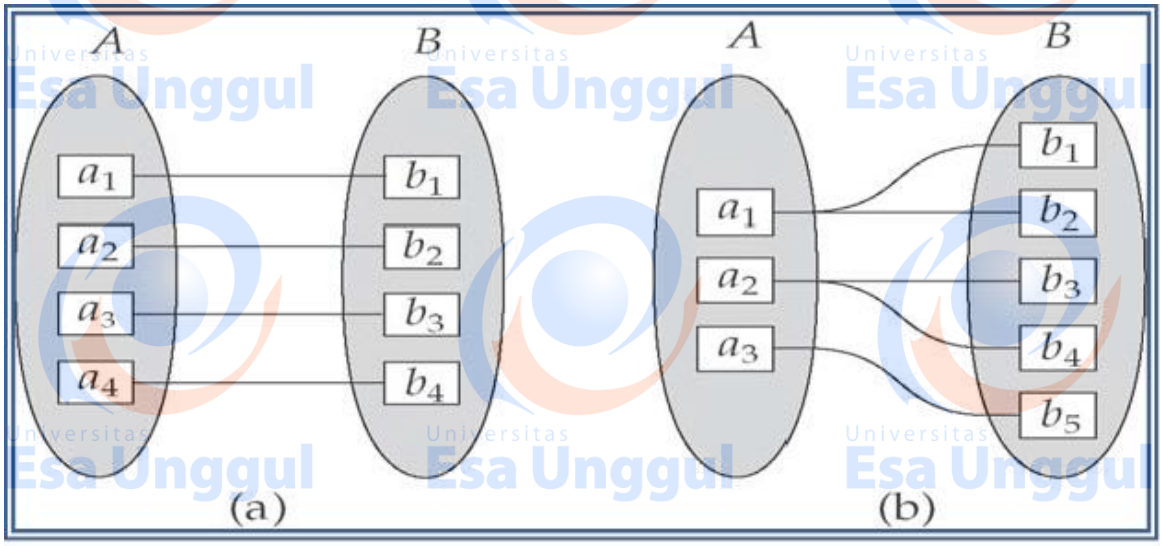
3. Relationship

   Relationship menggambarkan asosiasi (hubungan) yang nyata diantara beberapa entitas.

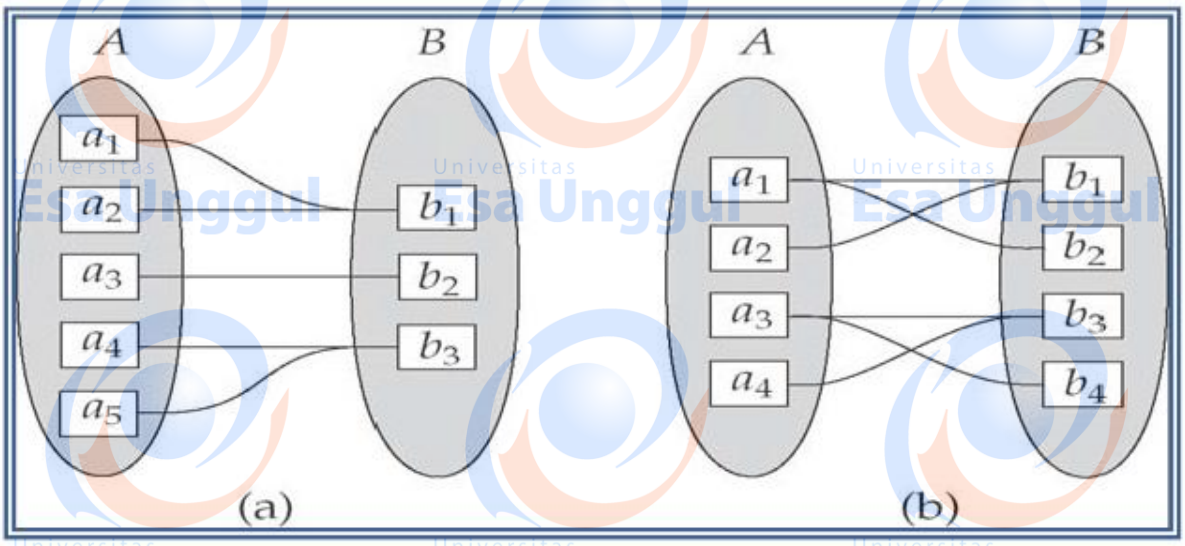
   Tingkatan dari relationship adalah:

   1. Unary: cuma ada satu entitas
   2. Binary: melibatkan dua entitas
   3. Ternary: melibatkan lebih dari dua entitas

Sedangkan derajat relasinya dapat one to one (Gambar 1.a), one to many (Gamba 1.b), many to one (Gambar 2.a) dan many to many (Gambar 2b).



Gambar 1 Derajat Relational



Gambar 2 Derajat Relational

Entity Relationship Diagram (ERD) pada Gambar 3 merupakan contoh basisdata sederhana dari perpustakaan. Dari ERD tersebut dapat dilihat bahwa hubungan antara entitas penerbit dan entitas buku adalah one to many, yang artinya satu penerbit dapat menerbitkan banyak buku. Hubungan antar entitas bisa terjadi karena adanya atribut yang berfungsi sebagai penghubung.Pada entitas penerbit dan buku terlihat ada kesamaan nama, yaitu kode_penerbit. Kode_penerbit digunakan sebagai penghubung antara kedua entitas.Atribut kode_penerbit pada tabel penerbit dinamakan primary key, sedangkan pada tabel buku dinamakan foreign key. Ciri-ciri primary key adalah:

1. Unik

   Tidak boleh terdapat data yang sama.
2. Not null

   Data harus ada untuk data yag diberikan nilai not null

Gambar 3 Contoh Basisdata Sederhana

Topik2

Tahap-Tahap Desain Database

Desain databse memegang peranan sangat penting dalam sistem database. Desain database ini selayaknya dilakukan setelah adanya analisis sistem dan mengetahui permasalahan ditempat database ini akan diterapkan. Sebuah database yang dibuat hanya berdasarkan insting atau intusi hasilnya tidak akan efisien. Karena bisa saja terjadi bahwa database yang dibuat tidak sanggup memenuhi permintaan konsumen. Bisa juga database tidak efisien dalam melakukan penyimpanan data, adanya redudansi atau data yang sama berulangkali dimasukan dalam tabel yang berbeda akan membuat database bekerja dengan lambat dan dengan cepat memenuhi ruang penyimpanan.

Tujuan Perancangan *Database*

a. Untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan *user* secara khusus dan aplikasi – aplikasinya.
b. Memudahkan pengertian struktur informasi.
c. Mendukung kebutuhan-kebutuhan pemrosesan dan beberapa obyek penampilan (*response time*, *processing time*, dan *storeage space*).

Proses perancangan *database* terdiri dari 6 tahap:

a. Tahap 1, Pengumpulan data dan analisis
b. Tahap 2, Perancangan database secara konseptual
c. Tahap 3, Pemilihan DBMS
d. Tahap 4, Perancangan *database* secara logika (*data model mapping*)
e. Tahap 5, Perancangan *database* secara fisik
f. Tahap 6, Implementasi Sistem *database*

Secara khusus proses perancangan berisi 2 aktifitas paralel:

1. Aktifitas yang melibatkan perancangan dari isi data dan struktur *database*,

2. Aktifitas mengenai perancangan pemrosesan *database* dan aplikasi-aplikasi perangkat lunak.

Di lain pihak, kita biasanya menentukan perancangan aplikasi-aplikasi *database* dengan mengarah kepada konstruksi skema *database* yang telah ditentukan selama aktifitas yang pertama.6 tahapan diatas tadi tidak harus diproses berurutan.Pada tahap ke 1 merupakan kumpulan informasi yang berhubungan dengan penggunaan database.Tahap 6 merupakan implementasi *database*-nya.Tahap 1 dan 6 kadang-kadang bukan merupakan bagian dari perancangan *database*. Sedangkan yang merupakan inti dari proses perancangan *database* adalah pada tahap 2, 4, 5.

a. Tahap 1 – Pengumpulan data dan analisa

Merupakan suatu tahap dimana kita melakukan proses indentifikasi dan analisa kebutuhan-kebutuhan data dan ini disebut pengumpulan data dan analisa. Untuk menentukan kebutuhan-kebutuhan suatu sistem *database*, kita harus mengenal terlebih dahulu bagian-bagian lain dari sistem informasi yang akan berinteraksi dengan sistem *database*, termasuk para *user* yang ada dan para *user*yang baru beserta aplikasi-aplikasinya. Kebutuhan-kebutuhan dari para *user* dan aplikasi-aplikasi inilah yang kemudian dikumpulkan dan dianalisa.

Berikut ini adalah aktifitas-aktifitas pengumpulan data dan analisa:

1. Menentukan kelompok pemakai dan bidang-bidang aplikasinya
2. Peninjauan dokumentasi yang ada
3. Analisa lingkungan operasi dan pemrosesan data
4. Daftar pertanyaan dan wawancara

b. Tahap 2, Perancangan *database* secara konseptual

Pada tahap ini akan dihasilkan *conceptual schema* untuk *database* yang tergantung pada sebuah DBMS yang spesifik. Sering menggunakan sebuah *high-level data model*seperti ER/EER modelselama tahap ini. Dalam *conceptual schema*, kita harus merinci aplikasi-aplikasi *database*yang diketahui dan

transaksi-transaksi yang mungkin.Tahap perancangan *database*secara konseptual mempunyai 2 aktifitas pararel:

1. Perancangan skema konseptual

   Menguji kebutuhan-kebutuhan data dari suatu *database* yang merupakan hasil dari tahap 1 dan menghasilkan sebuah *conceptual database schema* pada DBMS-*independent* model data tingkat tinggi seperti EER (*Enhanced Entity Relationship*) model.Untuk menghasilkan skema tersebut dapat dihasilkan dengan penggabungan bermacam-macam kebutuhan user dan secara langsung membuat skema database atau dengan merancang skema-skema yang terpisah dari kebutuhan tiap-tiap user dan kemudian menggabungkan skema-skema tersebut. Model data yang digunakan pada perancangan skema konseptual adalah DBMS-*independent* dan langkah selanjutnya adalah memilih DBMS untuk melakukan rancangan tersebut.

2. Perancangan transaksi

   Menguji aplikasi-aplikasi *database*dimana kebutuhan-kebutuhannya telah dianalisa pada fase 1, dan menghasilkan perincian transaksi-transaksi ini.Kegunaan tahap ini yang diproses secara paralel bersama tahapp perancangan skema konseptual adalah untuk merancang karakteristik dari transaksi-transaksi *database* yang telah diketahui pada suatu DBMS-*independent*. Transaksi-transaksi ini akan digunakan untuk memproses dan memanipulasi *database* suatu saat dimana *database* tersebut dilaksanakan.

c. Tahap 3, Pemilihan DBMS

   Pemilihan *database*ditentukan oleh beberapa faktor diantaranya faktor teknik, ekonomi, dan politik organisasi.

   Contoh faktor teknik:Keberadaan DBMS dalam menjalankan tugasnya seperti jenis-jenis DBMS (*relational, network, hierarchical*, dan lain-lain), struktur penyimpanan, dan jalur akses yang mendukung DBMS, pemakai, dan lain-lain.

   Faktor-faktor ekonomi dan organisasi yang mempengaruhi satu sama lain dalam pemilihan DBMS:

   1. Struktur data

Jika data yang disimpan dalam *database* mengikuti struktur hirarki, maka suatu jenis hirarki dari DBMS harus dipikirkan.

2. Personal yang telah terbiasa dengan suatu sistem

   Jika staf *programmer* dalam suatu organisasi sudah terbiasa dengan suatu DBMS, maka hal ini dapat mengurangi biaya latihan dan waktu belajar.

3. Tersedianya layanan penjual

   Keberadaan fasilitas pelayanan penjual sangat dibutuhkan untuk membantu memecahkan beberapa masalah sistem.

d. Tahap 4, Perancangan *database* secara logika (*data model mapping*)

Tahap selanjutnya adalah membuat sebuah skema konseptual dan skema eksternal pada model data dari DBMS yang terpilih.Tahap ini dilakukan oleh pemetaan skema konseptual dan skema eksternal yang dihasilkan pada tahap 2.Pada tahap ini, skema konseptual ditransformasikan dari model data tingkat tinggi yang digunakan pada tahap 2 ke dalam model data dari model data dari DBMS yang dipilih pada tahap 3.

Pemetaan tersebut dapat diproses dalam 2 tingkat:

1. Pemetaan *system-independent*

   Pemetaan ke dalam model data DBMS dengan tidak mempertimbangkan karakteristik atau hal-hal yang khusus yang berlaku pada implementasi DBMS dari model data tersebut.

2. Penyesuain skema ke DBMS yang spesifik

   Mengatur skema yang dihasilkan pada langkah 1 untuk disesuaikan pada implementasi yang khusus di masa yang akan datang dari suatu model data yang digunakan pada DBMS yang dipilih.Hasil dari tahap ini memakai perintah-perintah DDL (*Data Definition Language*) dalam bahasa DBMS yang dipilih yang menentukan tingkat skema konseptual dan eksternal dari sistem *database*.

   Tetapi 10 dalam beberapa hal, perintah-perintah DDL memasukkan parameter-parameter rancangan fisik sehingga DDL yang lengkap harus menunggu sampai tahap perancangan *database*secara fisik telah lengkap.Tahap ini dapat dimulai setelah pemilihan sebuah implementasi

model data sambil menunggu DBMS yang spesifik yang akan dipilih. Contoh: jika memutuskan untuk menggunakan beberapa *relational* DBMS tetapi belum memutuskan suatu relasi yang utama. Rancangan dari skema eksternal untuk aplikasi-aplikasi yang spesifik seringkali sudah selesai selama proses ini.

e. Tahap 5, Perancangan *database* secara fisik

Perancangan *database* secara fisik merupakan proses pemilihan struktur-struktur penyimpanan dan jalur-jalur akses pada *file-file database*untuk mencapai penampilan yang terbaik pada bermacam-macam aplikasi.Selama fase ini, dirancang spesifikasi-spesifikasi untuk *database* yang disimpan yang berhubungan dengan struktur-struktur penyimpanan fisik, penempatan record dan jalur akses. Berhubungan dengan *internal schema*(pada istilah 3 level arsitektur DBMS).

Beberapa petunjuk dalam pemilihan perancangan *database*secara fisik:

1. *Response time*

   Waktu yang telah berlalu dari suatu transaksi *database* yang diajukan untuk menjalankan suatu tanggapan. Pengaruh utama pada *response time* adalah di bawah pengawasan DBMS yaitu: waktu akses *database* untuk data item yang ditunjuk oleh suatu transaksi. *Response time* juga dipengaruhi oleh beberapa faktor yang tidak berada di bawah pengawasan DBMS, seperti penjadwalan sistem operasi atau penundaan komunikasi.

2. *Space utility*

   Jumlah ruang penyimpanan yang digunakan oleh *file-file database* dan struktur-struktur jalur akses.

3. *Transaction throughput*

   Rata-rata jumlah transaksi yang dapat diproses per menit oleh sistem *database*, dan merupakan parameter kritis dari sistem transaksi (misal: digunakan pada pemesanan tempat di pesawat, bank, dll). Hasil dari fase ini adalah penentual awal dari struktur penyimpanan dan jalur akses untuk *file-file database*.

f. Tahap 6, Implementasi Sistem *database*

Setelah perancangan secara logika dan secara fisik lengkap, kita dapat melaksanakan sistem *database*. Perintah-perintah dalam DDL dan SDL(*Storage Definition Language*) dari DBMS yang dipilih, dihimpun dan digunakan untuk membuat skema *database* dan *file-file database* (yang kosong). Sekarang *database*tersebut dimuat (disatukan) dengan datanya.Jika data harus dirubah dari sistem komputer sebelumnya, perubahan-perubahan yang rutin mungkin diperlukan untuk format ulang datanya yang kemudian dimasukkan ke *database* yang baru.

Transaksi-transaksi *database* sekarang harus dilaksanakan oleh para programmmer aplikasi.Spesifikasi secara konseptual diuji dan dihubungkan dengan kode program dengan perintah-perintah dari *embedded* DML yang telah ditulis dan diuji. Suatu saat transaksi-transaksi tersebut telah siap dan data telah dimasukkan ke dalam *database*, maka tahap perancangan dan implementasi telah selesai, dan kemudian tahap operasional dari sistem *database* dimulai.

Modul 2

Struktur Data Dengan Access

Modulpertemuan ini membahas tentang struktur data dengan menggunakan access.

Selain itu, secara khusus mahasiswa mampu untuk:

1. Menejelaskan cara membuat *database, create table* pada Ms. Access
2. Menjelaskan relational model antar data dan mengaplikasikan contoh kasus dengan menggunakan access
3. Menjelaskan konsep query
4. Membuat *design* data form, merancang laporan, *query processing,* dan membuat *switchboaord* dalam *database* access

Topik3

Relational Database

Database relasional adalah basis data digital berdasarkan model relasional data, seperti yang diusulkan oleh E. F. Codd pada tahun 1970. Sistem perangkat lunak yang digunakan untuk memelihara basis data relasional adalah sistem manajemen basis data relasional (RDBMS). Hampir semua sistem basis data relasional menggunakan SQL (Structured Query Language) untuk query dan pemeliharaan *database*.

Relational Model

Model ini mengatur data ke dalam satu atau beberapa tabel (atau "relasi") dari kolom dan baris, dengan kunci unik yang mengidentifikasi setiap baris. Baris juga disebut catatan atau tupel. Kolom juga disebut atribut. Umumnya, setiap tabel / relasi mewakili satu "tipe entitas" (seperti pelanggan atau produk). Baris mewakili contoh dari jenis entitas (seperti "Lee" atau "kursi") dan kolom yang mewakili nilai-nilai yang dikaitkan dengan contoh itu (seperti alamat atau harga).

Keys

Setiap baris dalam tabel memiliki kunci uniknya sendiri. Baris dalam tabel dapat dihubungkan ke baris di tabel lain dengan menambahkan kolom untuk kunci unik dari baris tertaut (kolom tersebut dikenal sebagai kunci asing). Codd menunjukkan bahwa hubungan data kompleksitas acak dapat diwakili oleh seperangkat konsep sederhana.

Bagian dari proses ini melibatkan secara konsisten dapat memilih atau memodifikasi satu dan hanya satu baris dalam sebuah tabel. Oleh karena itu, sebagian besar implementasi fisik memiliki kunci primer yang unik (PK) untuk setiap tabel.Ketika baris baru ditulis ke meja, nilai unik baru untuk kunci primer dihasilkan; ini adalah kunci yang digunakan sistem untuk mengakses tabel.Kinerja sistem dioptimalkan untuk PK.

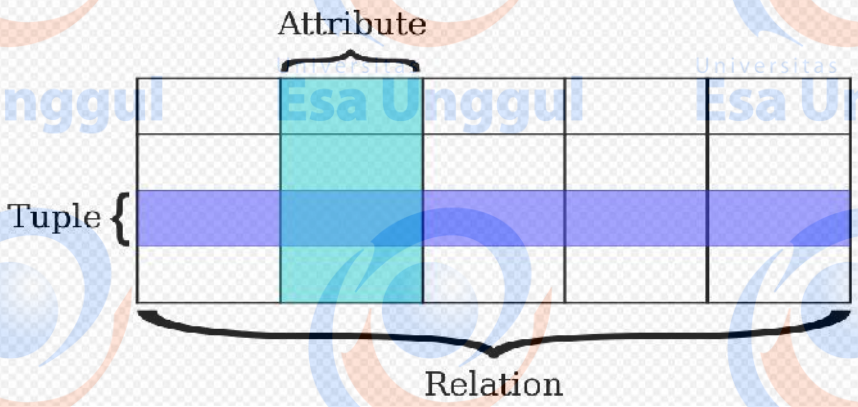
Kunci lainnya yang lebih alami juga dapat diidentifikasi dan didefinisikan sebagai tombol alternatif (AK).Seringkali beberapa kolom diperlukan untuk membentuk AK (ini adalah salah satu alasan mengapa kolom integer tunggal biasanya dibuat PK).Baik PK maupun AK memiliki kemampuan untuk mengidentifikasi baris secara unik dalam sebuah tabel.Teknologi tambahan dapat diterapkan untuk memastikan ID unik di seluruh dunia, pengenal unik global, ketika ada persyaratan sistem yang lebih luas.

Primary key dalam basis data digunakan untuk mendefinisikan hubungan antar tabel. Ketika seorang PK bermigrasi ke meja lain, itu menjadi kunci asing di meja lain. Ketika setiap sel hanya dapat berisi satu nilai dan PK bermigrasi ke dalam tabel entitas biasa, pola desain ini dapat mewakili hubungan satu-ke-satu atau satu-ke-banyak.

Sebagian besar desain basis data relasional menyelesaikan banyak hubungan banyak dengan membuat tabel tambahan yang berisi PK dari kedua tabel entitas lainnya hubungan tersebut menjadi sebuah entitas; tabel resolusi kemudian dinamai dengan tepat dan dua FK digabungkan untuk membentuk PK. Migrasi PK ke tabel lain adalah alasan utama kedua mengapa integer yang diberikan oleh sistem digunakan secara normal sebagai PK; jarang ada efisiensi atau kejelasan dalam memigrasikan sekelompok jenis kolom lainnya.

Relationships

Relationships adalah hubungan logis antara tabel yang berbeda, didirikan atas dasar interaksi di antara tabel-tabel.



Gambar 1 Terminologi *database relasional*

Database relasional pertama kali didefinisikan pada Juni 1970 oleh Edgar Codd, dari San Jose Research Laboratory IBM. Pandangan Codd tentang apa yang memenuhi syarat sebagai RDBMS diringkas dalam aturan 12 Codd. Database relasional telah menjadi tipe database yang dominan. Model lain selain model relasional termasuk model database hirarkis dan model jaringan.

Relations atau Tabel

Relasi didefinisikan sebagai kumpulan tupel yang memiliki atribut yang sama. Tuple biasanya mewakili objek dan informasi tentang objek itu.Objek biasanya berupa objek atau konsep fisik.Relasi biasanya digambarkan sebagai tabel, yang disusun dalam baris dan kolom. Semua data yang direferensikan oleh atribut berada dalam domain yang sama dan sesuai dengan batasan yang sama.

Model relasional menentukan bahwa tupel suatu relasi tidak memiliki urutan tertentu dan bahwa tupel, pada gilirannya, tidak memaksakan urutan pada atribut.Aplikasi mengakses data dengan menentukan kueri, yang menggunakan operasi seperti memilih untuk mengidentifikasi tupel, proyek untuk mengidentifikasi atribut, dan bergabung untuk menggabungkan hubungan.Hubungan dapat dimodifikasimenggunakan sisipan, hapus, dan perbarui operator.Tupel baru dapat memberikan nilai eksplisit atau diturunkan dari kueri.Demikian pula, pertanyaan mengidentifikasi tupel untuk memperbarui atau menghapus.

Tuples menurut definisinya unik.Jika tupel berisi kandidat atau primary key maka jelas itu unik; Namun, kunci primer tidak perlu didefinisikan untuk baris atau catatan untuk menjadi tuple.Definisi tuple mensyaratkan bahwa itu unik, tetapi tidak memerlukan Primary key untuk didefinisikan.Karena tuple itu unik, atributnya menurut definisi merupakan superkey.

Dasar dan Turunan dari Konsep Relasi

Dalam database relasional, semua data disimpan dan diakses melalui hubungan. Hubungan yang menyimpan data disebut "hubungan dasar", dan dalam implementasi disebut "tabel". Hubungan lain tidak menyimpan data, tetapi dihitung dengan menerapkan operasi relasional ke relasi lainnya. Hubungan ini kadang-kadang disebut "hubungan turunan". Dalam implementasi ini disebut "pandangan" atau "queri". Relasi turunannya nyaman karena mereka bertindak sebagai relasi

tunggal, meskipun mereka dapat mengambil informasi dari beberapa relasi. Juga, hubungan turunan dapat digunakan sebagai lapisan abstraksi.

Domain

Domain menjelaskan kumpulan nilai yang mungkin untuk atribut yang diberikan, dan dapat dianggap sebagai kendala pada nilai atribut. Secara matematis, melampirkan domain ke atribut berarti bahwa nilai apa pun untuk atribut harus berupa elemen dari kumpulan yang ditentukan. String karakter "ABC", misalnya, tidak dalam domain bilangan bulat, tetapi nilai integer 123 adalah. Contoh lain dari domain menjelaskan nilai yang mungkin untuk bidang "CoinFace" sebagai ("Heads", "Tails"). Jadi, bidang "CoinFace" tidak akan menerima nilai input seperti (0,1) atau (H, T).

Constraints

Batasan memungkinkan untuk lebih membatasi domain dari suatu atribut. Misalnya, kendala dapat membatasi atribut integer yang diberikan ke nilai antara 1 dan 10. Batasan menyediakan satu metode untuk menerapkan aturan bisnis dalam database dan mendukung penggunaan data berikutnya dalam lapisan aplikasi.

SQL mengimplementasikan fungsi kendala dalam bentuk cek kendala. Batasan membatasi data yang dapat disimpan dalam hubungan. Ini biasanya didefinisikan menggunakan ekspresi yang menghasilkan nilai boolean, menunjukkan apakah data memenuhi kendala atau tidak. Batasan dapat berlaku untuk atribut tunggal, ke tuple (membatasi kombinasi atribut) atau ke seluruh relasi. Karena setiap atribut memiliki domain yang terkait, ada kendala (kendala domain). Dua aturan utama untuk model relasional dikenal sebagai integritas entitas dan integritas referensial.

Primary Key

Primary key secara unik menentukan tupel dalam tabel. Primary key yang baik, ia tidak harus mengulangi data yang sama. Sementara atribut alami (atribut yang digunakan untuk mendeskripsikan data yang dimasukkan) terkadang merupakan kunci primer yang baik, kunci pengganti sering digunakan sebagai gantinya.

Kunci pengganti adalah atribut buatan yang diberikan kepada objek yang secara unik mengidentifikasi itu (misalnya, dalam tabel informasi tentang siswa di sekolah mereka mungkin semua diberi ID siswa untuk membedakan mereka).

Kunci pengganti tidak memiliki makna intrinsik (inheren), tetapi lebih berguna melalui kemampuannya untuk mengidentifikasi tuple secara unik. Kejadian umum lainnya, terutama dalam hal kardinalitas N:M adalah kunci komposit.

Kunci komposit adalah kunci yang terdiri dari dua atau lebih atribut dalam tabel yang (bersama-sama) secara unik mengidentifikasi suatu catatan. (Misalnya, dalam database yang menghubungkan siswa, guru, dan kelas. Kelas dapat diidentifikasi secara unik dengan kunci gabungan dari nomor kamar dan slot waktu mereka, karena tidak ada kelas lain yang dapat memiliki kombinasi atribut yang sama. Bahkan, penggunaan kunci komposit seperti ini dapat menjadi bentuk verifikasi data, meskipun yang lemah.

Foreign Key

Foreign key adalah bidang dalam tabel relasional yang cocok dengan kolom primary key dari tabel lain. *Foreign key* dapat digunakan untuk tabel referensi silang.*Foreign key* tidak perlu memiliki nilai unik dalam relasi referensi.*Foreign key* secara efektif menggunakan nilai-nilai atribut dalam relasi yang direferensikan untuk membatasi domain dari satu atau lebih atribut dalam relasi referensi.

Foreign key dapat dideskripsikan secara formal sebagai: "Untuk semua tupel dalam relasi referensi yang diproyeksikan di atas atribut referensi, harus ada tuple dalam relasi yang direferensikan yang diproyeksikan atas atribut-atribut yang sama sehingga nilai-nilai dalam masing-masing atribut referensi sesuai dengan yang bersangkutan. Nilai-nilai dalam atribut yang direferensikan.

Topik4

Struktur Data Dengan Access (Kasus: Database Perpustakaan)

Langkah-langkah Membuat Database Perpustakaan

Create tabel buku

Tabel : Buku		
Field Name	**Data Type**	**Keterangan**
no_buku	Text (4)	Primary key
judul	Text (100)	
Pengarang	Text (40)	Dengan asumsi, nama pengarang lebih dari satu pengarang
tahun_terbit	Number (integer)	
Jenis_buku	Text (30)	
Status	Text (20)	Tersedia, Dipinjam, Hilang, Rusak

Create tabel anggota

Tabel : Anggota		
Field Name	**Data Type**	**Keterangan**
id_anggota	Text (6)	Primary key
nama	Text (40)	
alamat	Text (50)	
kota	Text (25)	
no_telp	Text (15)	
tanggal_lahir	Date/Time	Format shotdate

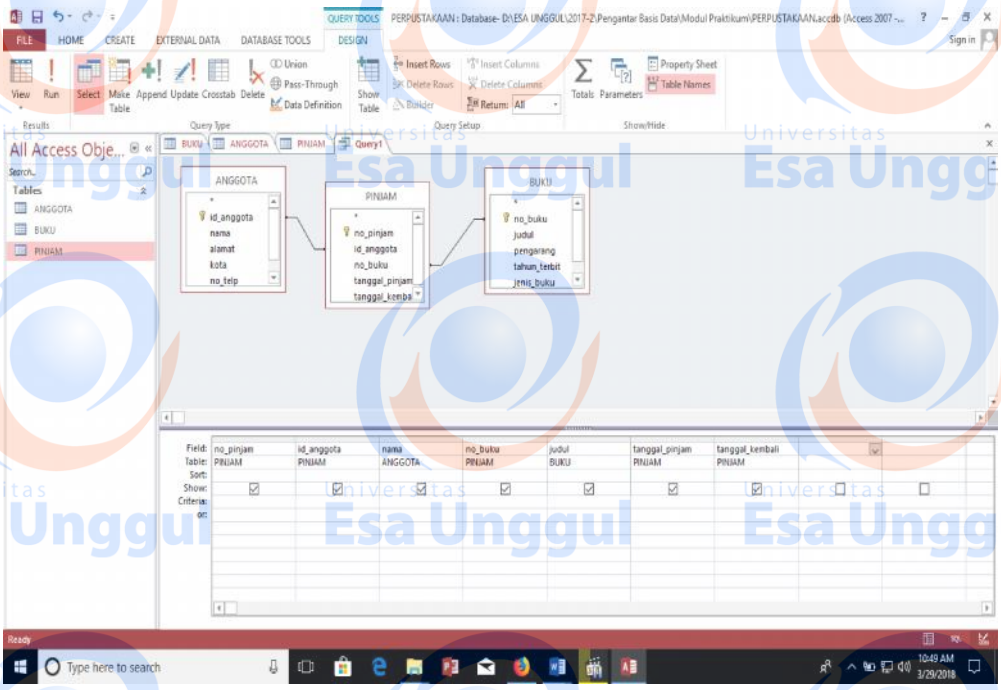
Create tabel pinjam

Tabel : Pinjam		
Field Name	**Data Type**	**Keterangan**
no_pinjam	Text (5)	Primary key
Id_anggota	Text (6)	
no_buku	Text (4)	
tanggal_pinjam	Date/Time	Format shotdate
tanggal_kembali	Date/Time	Format shotdate

Setiap tabel diisi dengan 10 data.

Create Query

Pilih query design, maka muncul beberapa tabel yang telah di create, seperti terlihat pada gambar dibawah ini:



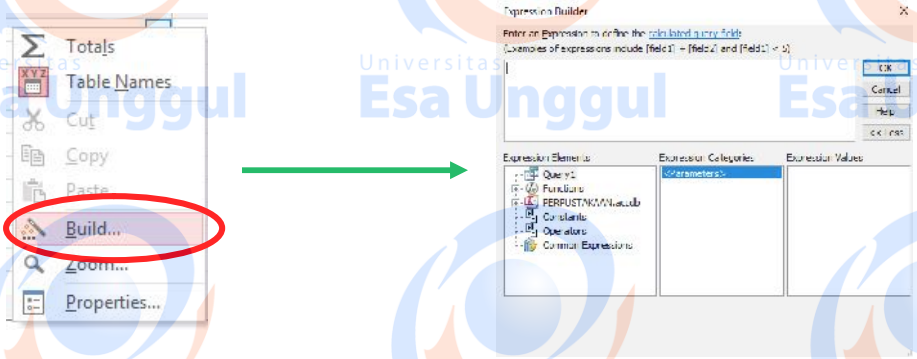
Pada latihan ini, kita ingin menampilkan:

a. No_pinjam dari tabel pinjam
b. Id_anggota dari tabel pinjam

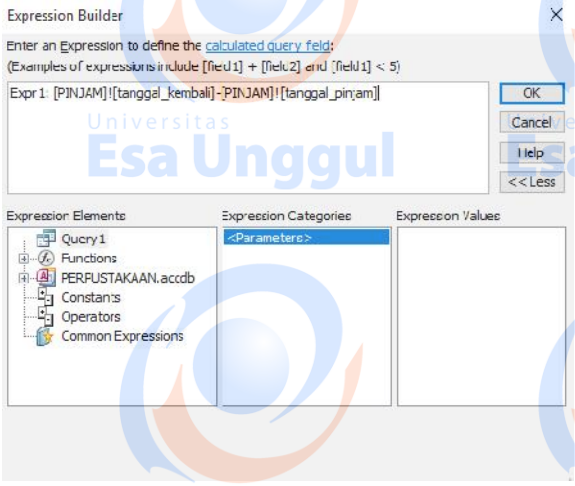
c. Nama anggota dari tabel anggota
d. No_buku dari tabel pinjam
e. Judul buku dari tabel buku
f. Tanggal_pinjam dari tabel pinjam
g. Tanggal_kembali dari tabel pinjam
h. Lama pinjam

Untuk lama pinjam di-create dengan cara

a. Ceklis kotok yang terdapat pada baris kategori show
b. Klik kanan pada baris yang kosong (row setelah tanggal_kembali)
c. Klik build



d. Untuk membuat lama pinjam adalah **tanggal kembali – tanggal pinjam**, dengan cara klik database perpustakaan yang terdapat pada kolom **Expression Elements → Pilih Tabel → Pilih tabel pinjam** sehingga munculfield-field yang terdapat pada tabel pinjam (**Expression Categories**). Double klik tanggal_kembali → klik operator (-) → klik tanggal pinjam.



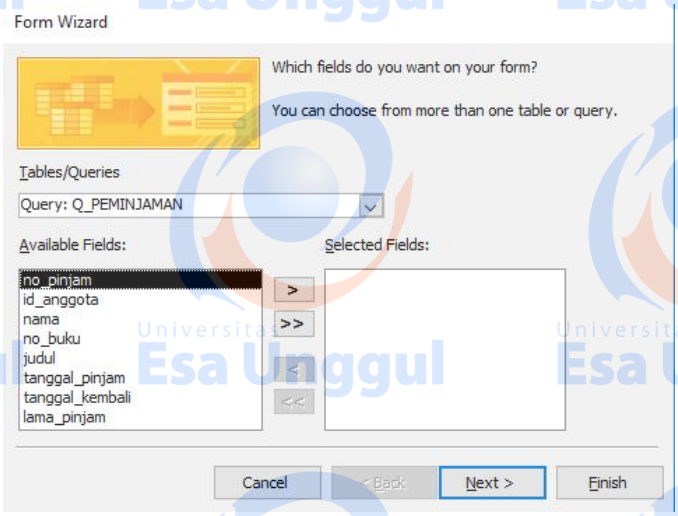
e. Klik OK dan simpan dengan nama Q_PEMINJAMAN

f. Ganti nama Expr1 dengan Lama_pinjam

Create Form

Form dibuat dengan menggunakan form wizard, dengan cara:

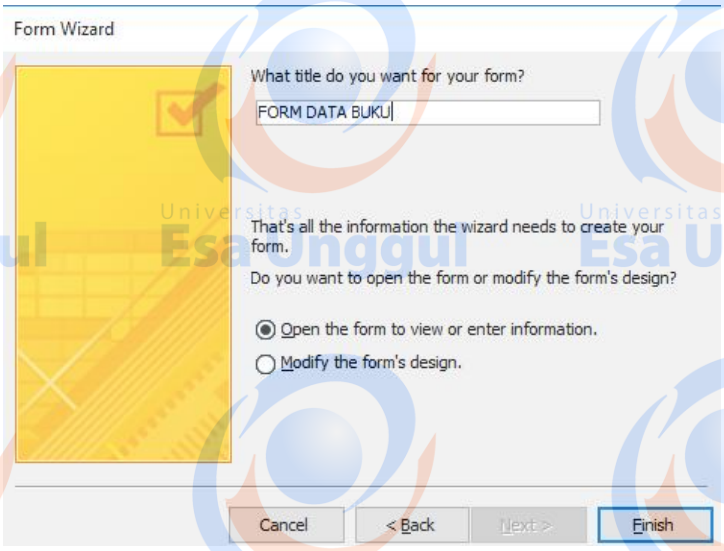
a. Klik form wizard



b. Pilih tabel buku

c. Pindahkan seluruh field yang terdapat pada **available fields** ke **Selected Fileds** dengan cara klik tombol (**>>**)

d. Klik next sampai menghasilkan tampilan seperti gambar dibawah ini

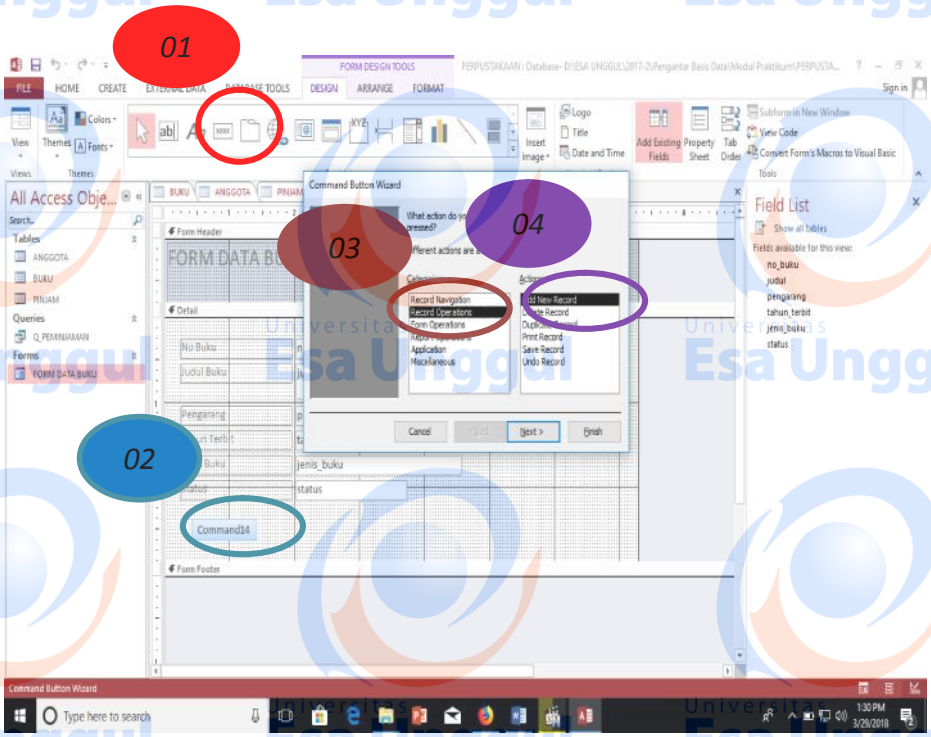


e. Ganti title dengan "**FORM DATA BUKU**", Finish

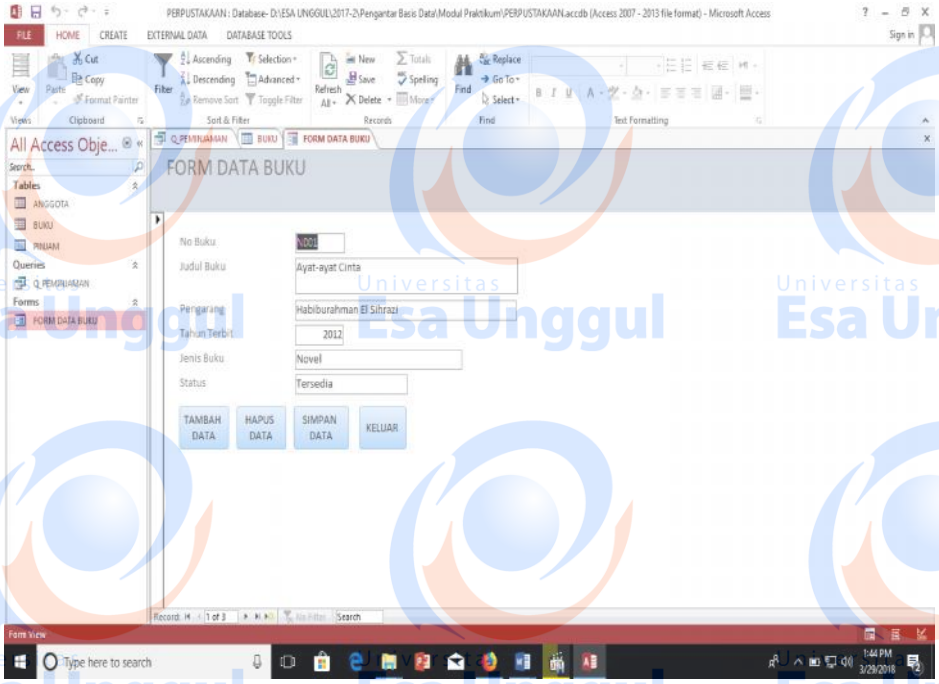
1. Menambahkan tombol

   a. Tombol tambah

      Klik button, arahkan ke form data buku, seperti gambar dibawah ini:

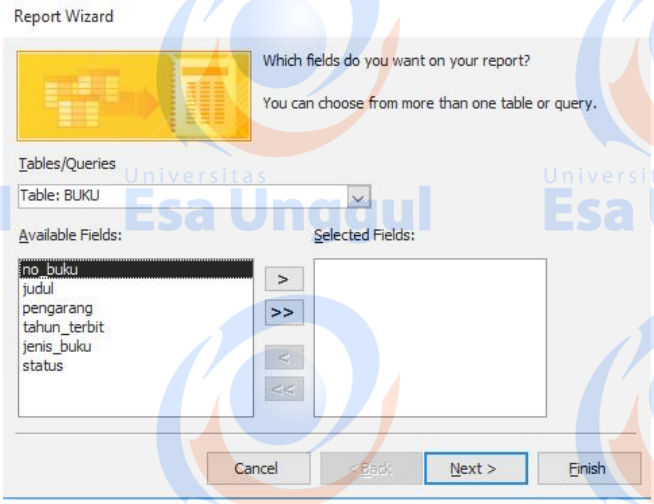


   b. Create untuk button hapus data, simpan data, dan keluar dengan cara yang sama seperti perintah 7a, sehingga desain form seperti gambar dibawah.

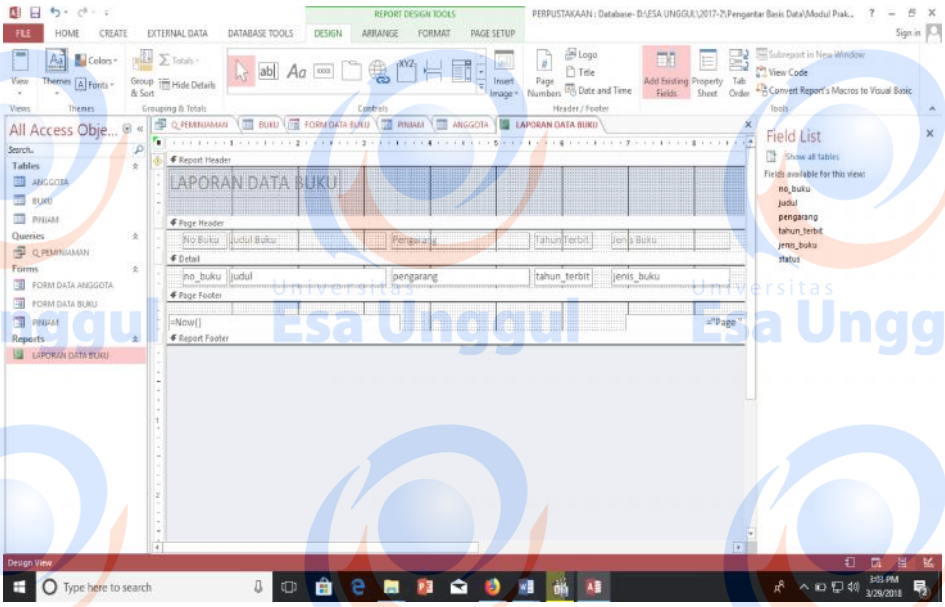


Membuat Laporan

1. Klik report wizard, seperti yang terlihat pada gambar dibawah.



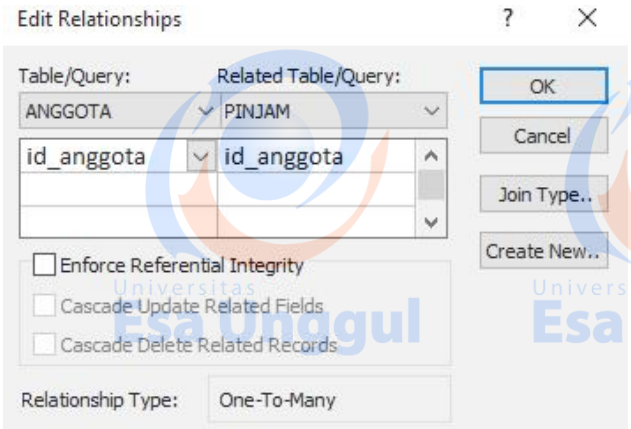
2. Pada tables/query pilih tabel yang diinginkan untuk proses pembuatan laporan.



3. Selesai

Membuat Relasi

1. Klik database tools
2. Klik relationship
3. Hubungkan/relasikan antara primary key dengan foreign key
4. Untuk menentukan jenis relasi maka double klik garis relasi sehingga tampil seperti gambar dibawah:

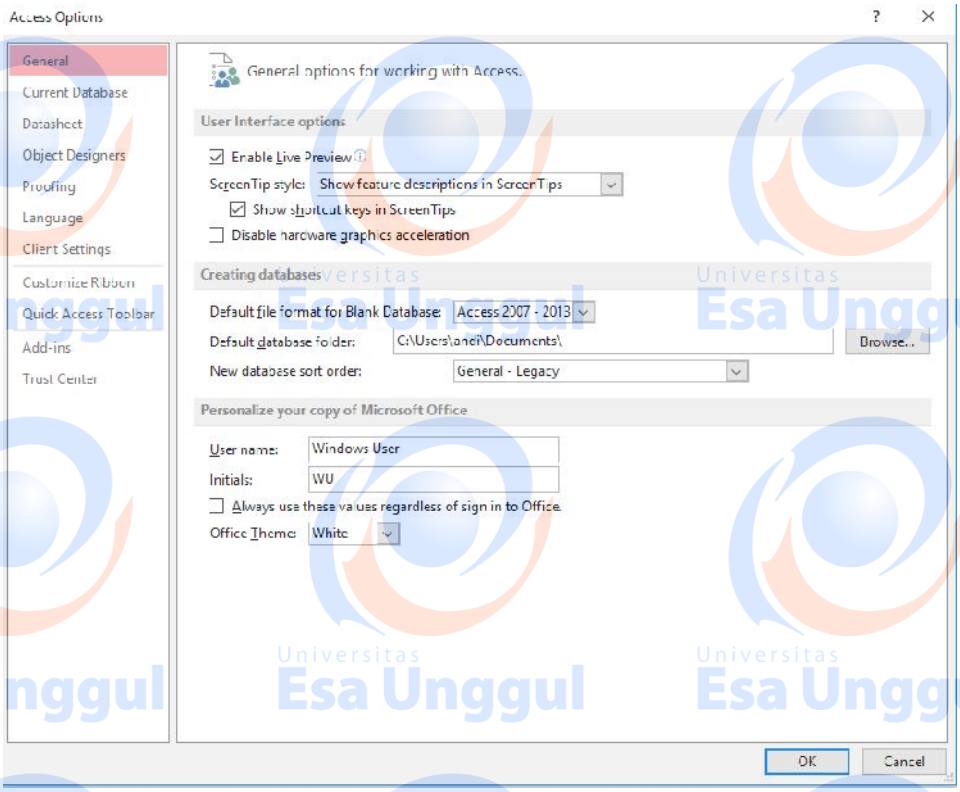


5. Ceklist enforce referential integrity, cascade update related fields, dan cascade delete related records
6. Ok

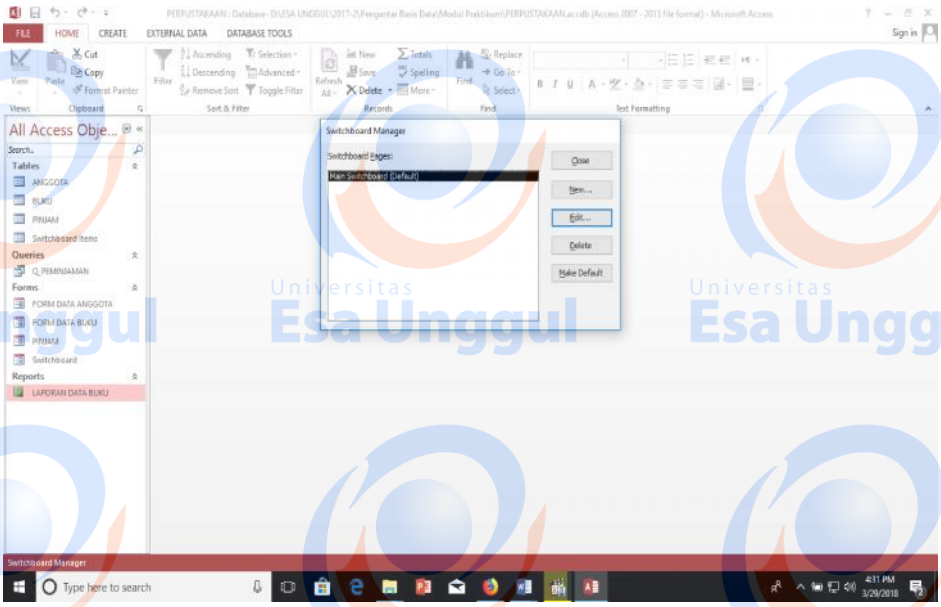
Membuat Menu Utama

Dalam membuat menu utama, maka langkah awal adalah mengklik switchboard manager. Pada Microsoft office 2010 sampai versi terbaru switchboard manager tidak ditampilkan. Cara menampilkan switchboard manager pada menu Microsoft acces tersebut adalah:

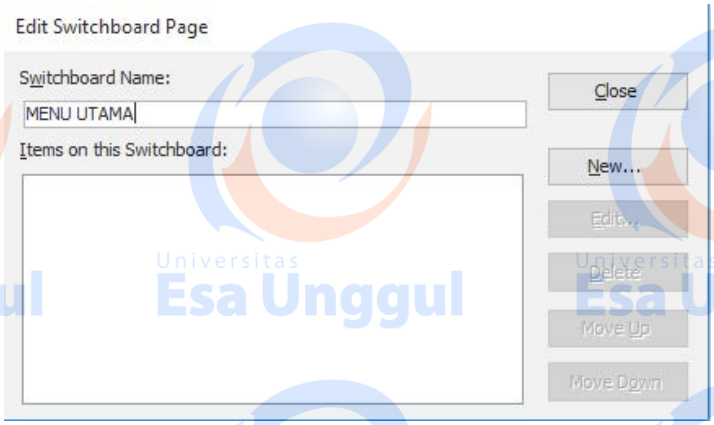
1. Klik file
2. Klik option



3. Klik customize → pada choose commands from, pilih all comment
4. Pilih switchboard manager dan klik tombol add
5. Setelah switchboard manager ditambahkan pada menu Ms. Access, klik switchboard manager sehingga muncul seperti gambar berikut:



6. Klik edit



7. Tambahkan menu yang mau dimunculkan dalam switchboard page, dengan cara:

a. Klik New, sehingga tampil edit switchboard item

b. Ganti text sesuai dengan menu yang ingin ditampilkan

Contoh: Menu data buku

Pada Text : Data Buku

Pada Command : Open Form in Edit Mode

Pada Switchboard : Form Data Buku

c. Close
d. Save as
e. Pilih Make ACCDE

Latihan 2.1

1. Create form tabel anggota
2. Create form tabel peminjaman, untuk filed id anggota dan no buku gunakan tool combo box.
3. Create laporan tabel anggota
4. Create laporan tabel pinjam
5. Create semua menu form dan menu laporan pada switchboard manager

Modul 3

Struktur Data Dengan MySQL

Modul pada pertemuan ini, menjelaskan tentang MySQL, yaitu:

1. Cara menggunakan tool MySQL manager.
2. Cara memanipulasi objek-objek database.
3. Memanipulasi data-data dalam database menggunakan kode SQL.
4. Menggunakan dan menerapkan fungsi-fungsi MySQL (Built in Function)

Topik5

MySQL

MySQL merupakan sebuah RDBMS yang berbasiskan database server.Kemampuan MySQL dalam menangani RDBMS mengakibatkan MySQL menjadi sangat terkenal dan banyak digunakan saat ini.MySQL mampu menangani data yang sangat besar sehingga cocok untuk menangani data pada perusahaan besar maupun kecil.

Tidak seperti PHP atau Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. MySQL dapat didownload di situs resminya, http://www.mysql.com.

Kelebihan MySQL

Database MySQL memiliki beberapa kelebihan dibanding database lain, diantaranya:

- MySQL merupakan Database Management System (DBMS)
- MySQL sebagai Relation Database Management System (RDBMS) atau disebut dengan database Relational
- MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya
- MySQL merupakan sebuah database client
- MySQL mampu menerima query yang bertupuk dalam satu permintaan atau Multi-Threading.
- MySQL merupakan Database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran GigaByte sekalipun.

- MySQL diidukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti visual Basic dan Delphi.
- MySQL adalah database menggunakan enkripsi password, jadi database ini cukup aman karena memiliki password untuk mengakses nya.
- MySQL merupakan Database Server yang multi user, artinya database ini tidak hanya digunakan oleh satu pihak orang akan tetapi dapat digunakan oleh banyak pengguna.
- MySQL mendukung field yang dijadikan sebagai kunci primer dan kunci uniq (Unique).
- MySQL memliki kecepatan dalam pembuatan table maupun peng-update an table.

Mengenal SQL (Structured Query Language)

SQL (Structured Query Language) adalah sebuah bahasa permintaan database yang terstruktur.Bahasa SQL ini dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam database maupun merelasikan antar database.SQL dibagi menjadi tiga bentuk Query, yaitu:

1. DDL (*Data Definition Language*)
   DDL adalah sebuah metode Query SQL yang berguna untuk mendefinisikan data padasebuah Database, Query yang dimiliki DDL adalah:
   - CREATE : Digunakan untuk membuat Database dan Tabel
   - Drop : Digunakan untuk menghapus Tabel dan Database
   - Alter : Digunakan untuk melakukan perubahan struktur tabel yang telahdibuat, baik menambah Field (Add), mengganti nama Field(Change) ataupun menamakannya kembali (Rename), danmenghapus Field (Drop).
2. DML (*Data Manipulation Language*)
   DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi,sehingga fungsi dari Query DML ini untuk melakukan pemanipulasian database yang telah dibuat. Query yang dimiliki DML adalah:
   - INSERT : Digunakan untuk memasukkan data pada Tabel

Database

- UPDATE : Digunakan untuk pengubahan terhadap data yang ada pada Tabel Database
- DELETE : Digunakan untuk Penhapusan data pada tabel Database

3. DCL (*Data Control Language*)

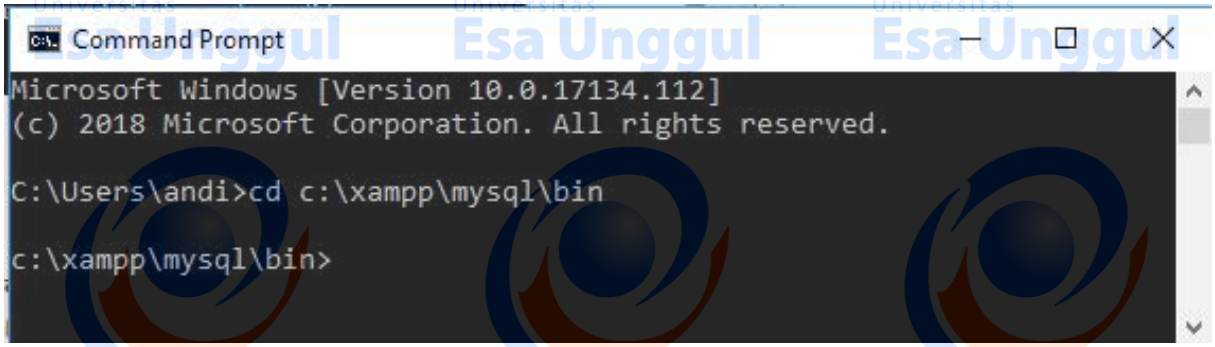
DCL adalah sebuah metode Query SQL yang digunakan untuk memberikan hakotorisasi mengakses Database, mengalokasikan space, pendefinisian space, dan pengauditan penggunaan database. Query yang dimiliki DCL adalah:

- GRANT : Untuk mengizinkan User mengakses Tabel dalam Database.
- REVOKE GRANT : Untuk membatalkan izin hak user, yang ditetapkan oleh perintah
- COMMIT : Mentapkan penyimpanan Database
- ROLLBACK : Membatalkan penyimpanan Database

Mengaktifkan Direktori MySQL Server

Untuk dapat menggunakan MySQL, terlebih dahulu aktifkan server MySQL dengan menghidupkan daemond MySQL. Program MySQL yang digunakan adalah XAMPP, maka untuk menjalankan daemond MySQL terdapat pada direktori yaitu C:\Program Files\Xampp\Mysql\Bin.

Untuk masuk kedalam server MySQL, bukalah MS-DOS Prompt anda melalui **Run** kemudian**ketik Command atau cmd**. Maka anda dapat masuk ke dalam direktori MySQL melalui MS-DOS Promtpseperti dibawah ini.



Masuk dan Keluar dari Server MySQL

MySQL adalah sebuah database server yang sangat aman.MySQL memiliki kemampuanmemanajemen user dalam mengakses.Jadi, tidak sembarang user dapat mengakses sebuah databaseyang diciptakan MySQL. Maka sebelum anda memiliki User untuk mengakses MySQL anda juga dapatMengakses database MySQL menggunakan User *Root.*

Berikut adalah perintah yang digunakan untuk mengkoneksikan kedalam server MySQL:

Shell > MySQL –u Root –p

Enter Password: *********

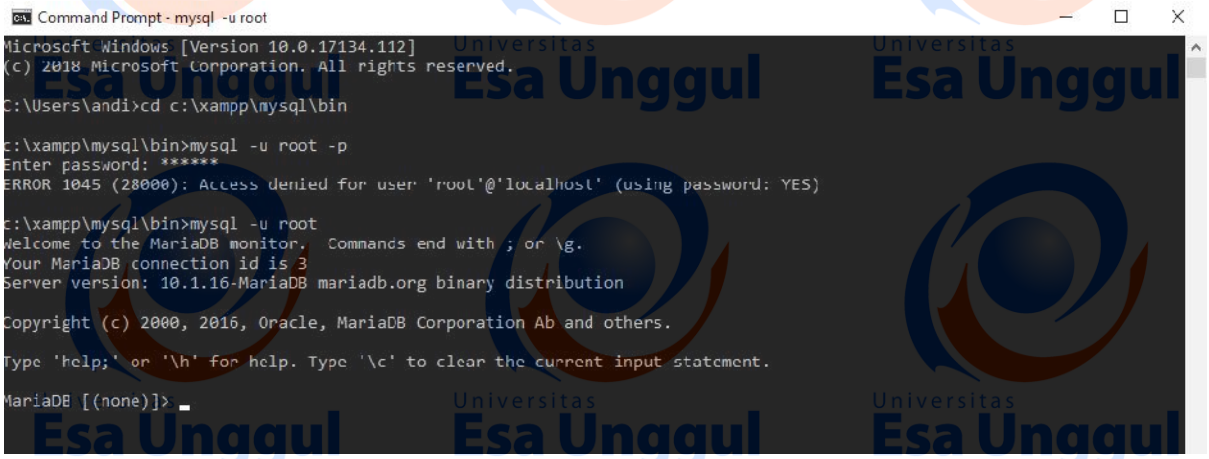
Keterangan:

Tanda –u menerangkan bahwa kita akan masuk menggunakan User Name bernamaRoot.

Tanda *–p* menyatakan kita akan masuk menggunakan Password.

Berikut adalah perintah yang digunakan untuk mengkoneksikan kedalam Server Mysql melalui Root:

Shell> Mysql –u root



Untuk dapat keluar dari Server MysQL kita dapat mengetikkan Intruksi quit ata \q:

Mysql> quit

Bye

Mysql> \q

Bye

DAFTAR PUSTAKA

Heryanto, I. 2017. Membuat Database dengan MS. ACCESS, Studi Kasus: Sistem Informasi Kepegawaian. ISBN: 978-602-6232-38-0.

Saputro H. 2012. Modul Pembelajaran Praktek Basis Data (MySQL)