BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian sebelumnya yang dilakukan oleh penulis lain yang dijadikan sebagai bahan pertimbangan yang berkaitan dengan penelitian yang pertama adalah "Aplikasi *E-learning* untuk Bimbingan Belajar Kumon" Penelitian serupa dilakukan pada tahun 2018. Aplikasi ini bertujuan untuk mempermudah siswa dalam mendapatkan materi dan berinteraksi dengan guru, serta dapat membantu orang tua siswa agar bisa mendapatkan informasi tentang perkembangan anaknya. Metode penelitian yang digunakan adalah metode observasi dan metode wawancara serta studi pustaka. Sedangkan metode pengembangan sistemnya menggunakan metode *waterfall*[2].

Penelitian serupa juga dilakukan pada tahun 2018 dengan judul "Pengembangan Aplikasi *E-learning* pada Bimbingan Belajar Kiswah" aplikasi ini dibuat untuk mempermudah siswa baru dalam melakukan pendaftaran. Lalu tujuan lain dari aplikasi ini yaitu supaya siswa bisa menerima tugas dari mentor yang sudah meng*upload* tugasnya tersebut. Metode pengumpulan data yang digunakan adalah studi literatur sejenis, studi pustaka, dan wawancara. Sedangkan metode pengembangan sistem menggunakan RAD (Rapid Application Development) Aplikasi ini dibangun dengan bahasa pemrogaman PHP dan framework Codeigniter serta MySql sebagai databasenya[3].

Penelitian serupa lainnya juga dilakukan dengan judul "Sistem Informasi Pembelajaran *Online* pada Bimbingan Belajar Cyber Solution" dilakukan pada tahun 2021. Sistem ini mampu memberikan atau menyediakan sarana belajar mengajar tanpa harus melalui tatap muka secara langsung. Metode pengembangan perangkat lunak yang digunakan adalah dengan menggunakan metode

waterfall. Sedangkan metode pengumpulan data yang digunakan adalah studi literatur, studi pustaka, dan wawancara[4].

Terakhir, adapun penelitian internasional yang dilakukan pada tahun 2021 dengan judul "The university students' viewpoints on e-learning system during COVID-19 pandemic: the case of Iran" yang berarti "Pandangan mahasiswa tentang sistem e-learning selama COVID-19 pandemi: kasus Iran". Tujuan dari penelitian ini adalah untuk menjelaskan kekuatan dan kelemahan sistem *e-learning* di universitas ilmu kedokteran Iran dalam COVID-19 pandemi. Sistem *e-learning* adalah sebuah alat penting untuk melanjutkan pendidikan selama pandemi COVID-19. Sebagian besar siswa percaya bahwa *e-learning* adalah pelengkap yang bagus untuk mencegah kegagalan akademik, tetapi tidak dapat meniru efisiensi yang sama dari pelatihan tatap muka[5].

Berbeda dengan penelitian-penelitian terkait yang dilakukan sebelumnya. Adapun penelitian yang dikembangkan berjudul "Pengembangan *E-learning* Di Bimbel Nedsela Study Club Menggunakan *Website*" dengan study kasus di Bimbel Nedsela Study Club. Sistem ini dapat dioperasikan oleh beberapa aktor diantaranya yaitu admin, mentor, siswa, dan orang tua siswa. Sistem ini bertujuan untuk memudahkan siswa dalam pembelajaran jarak jauh apabila mentor berhalangan hadir atau pada saat kelas pengganti, memudahkan mentor dan orang tua untuk memantau siswa serta memudahkan siswa untuk mempelajari materi-materi yang telah disampaikan sebelumnya oleh mentor.

2.2 Landasan Teori

Landasan teori berisi hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup permasalahan sebagai landasan dalam pembuatan laporan ini.

2.2.1 *E-Learning*

E-Learning terdiri dari dua bagian yaitu 'e' yang merupakan singkatan dari '*electronic*' dan '*learning*' yang berarti 'pembelajaran'. Jadi, *e-learning* berati pembelajaran dengan menggunakan jasa bantuan perangkat elektronika, khususnya perangkat komputer. Karena itu, maka *e-learning* sering disebut pula

dengan *'online course'*[6]. Dalam berbagai literatur, *e-learning* didefinisikan sebagai berikut : *e-learning* atau pembelajaran melalui *online* adalah pembelajaran yang pelaksanaannya didukung oleh jasa teknologi seperti telepon, audio, *vidiotape*, transmisi satelit atau komputer[7].

2.2.2 Lembaga Bimbingan Belajar

Bimbingan belajar adalah bimbingan yang ditujukan kepada siswa untuk mendapat pendidikan yang sesuai dengan kebutuhan, bakat, minat, kemampuannya dan membantu siswa untuk menentukan cara-cara yang efektif dan efisien dalam mengatasi masalah belajar yang dialami oleh siswa. Sedangkan pengertian lainnya tentang bimbingan belajar yaitu proses pemberian bantuan kepada murid dalam memecahkan kesulitan-kesulitan yang berhubungan dengan masalah belajar. Berdasarkan pengertian diatas bimbingan belajar dapat diartikan suatu proses pemberian bantuan kepada siswa dalam menyelesaikan masalah-masalah belajar yang dihadapi siswa, sehingga tercapai tujuan belajar yang diinginkan[8].

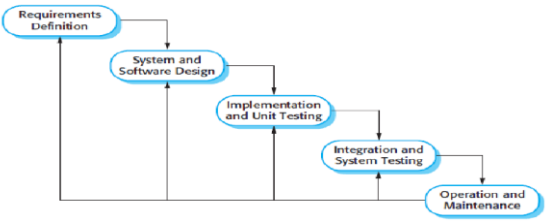
2.2.3 PHP *Mailer*

PHP *Mailer* merupakan salah satu *library* yang disediakan dan bersifat *free*. *Library* ini dapat dijalankan di berbagai macam *framework* PHP, *Library* ini memungkingkan *user* dapat memanfaatkan *e-mail* sebagai media interaksi antara *user* dengan sistem.

PHP *Mailer* adalah salah satu *library* PHP *open source* yang digunakan untuk mengirim *e-mail* dari *localhost*. PHP *Mailer* dapat menjalankan fungsinya sebagai *e-mail* jika kita men*support*nya dengan *Simple Mail Transfer Protocol* (SMTP). SMTP adalah suatu *protocol* yang diperlukan untuk mengirim dan menerima *e-mail*. Karena itulah kita harus menggunakan SMTP sebagai layanan mengirim *e-mail*. Layanan ini dapat digunakan untuk keperluan seperti menverifikasi *e-mail*, contohnya ketika mendaftarkan di Twitter atau Facebook. Setelah mendaftar maka kita diharuskan membuka *e-mail* dan memverifikasinya[9]. PHP *Mailer* dapat digunakan sebagai notifikasi pemberitahuan kepada orang tua siswa mengenai jadwal bimbel untuk melakukan bimbel .

2.2.4 Metode *Waterfall*

Metode *Waterfall* adalah sebuah metode pengembangan sistem dimana antara satu fase ke fase yang lain dilakukan secara berurutan. Dalam proses implementasi metode *waterfall* ini, sebuah langkah akan diselesaikan terlebih dahulu dimulai dari tahapan yang pertama sebelum melanjutkan ke tahapan yang berikutnya. Adapun keuntungan menggunakan metode *waterfall* yaitu *requirement* harus didefinisikan lebih mendalam sebelum proses *coding* dilakukan, selain itu, proses implementasinya dilakukan secara bertahap dari tahap pertama hingga tahap terakhir secara berurutan. Disamping itu, metode *waterfall* juga memungkinkan sedikit mungkin perubahan yang dilakukan oleh proyek yang sedang berlangsung[10]. Adapun metode *waterfall* menurut Sommerville menjelaskan bahwa metode *waterfall* memiliki tahapan utama dari *waterfall model* yang mencerminkan aktifitas pengembangan dasar. Terdapat 5 (lima) tahapan pada metode waterfall, yaitu *requirement analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance*.



Gambar 2. 1 Metode *Waterfall* Sommerfile

Adapun tahap-tahapanya yaitu :

a. *Requirements Definition* (Analisa Kebutuhan Perangkat Lunak) adalah tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak

seperti apa yang dibutuhkan oleh *user*. Melakukan wawancara dengan pemilik bimbel Nedsela *Study Club* untuk mengetahui sistem pembelajaran yang sudah berjalan seperti apa, kendala yang muncul selama proses pembelajaran apa saja serta mempertanyakan kebutuhan *user*. Proses belajar mengajar pada Bimbel Nedsela *Study Club* ini membutuhkan suatu sistem terkomputerisasi untuk mempermudah proses belajar mengajar jarak jauh.

b. *System and Software Design* (Desain atau Perancangan Sistem)
Proses perancangan sistem dibagi menjadi 2 bagian yaitu perancangan sistem perangkat keras dan perangkat lunak. Pada tahap ini, hasil analisa kebutuhan perangkat lunak pada sistem ini akan dideskripsikan ke dalam beberapa diagram, antara lain *use case diagram*, *sequence diagram*, dan *class diagram* yang pembuatannya menggunakan software microsoft visio dan visual paradigm.

c. *Implementation and Unit Testing* (Penulisan Kode Program)
Desain atau perancangan sistem harus ditranslasikan ke dalam program perangkat lunak. Pada tahap ini, apa yang sudah didapatkan pada saat analisis kebutuhan dan hasil desain diimplementasikan ke dalam kode atau bahasa yang dimengerti oleh komputer agar program komputer sesuai dengan hasil desain yang telah dibuat.

d. *Integration and Testing System* (Pengujian)
Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Sistem diuji secara menyeluruh sebagai sebuah sistem lengkap untuk memastikan apakah sesuai kebutuhan atau tidak. Setelah pengujian selesai dilakukan, perangkat lunak dapat dikirimkan ke Bimbel Nedsela *Study Club* untuk digunakan sebagaimana mestinya

e. *Operation and Maintenance* (Pemeliharaan)

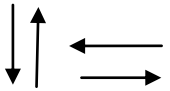
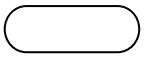
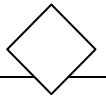
Pada tahapan ini, program dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak di temukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai suatu kebutuhan sistem yang baru.

2.2.5 *Flowchart*

Flowchart adalah bagan (*chart*) yang menunjukan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi[11].

Simbol-simbol dalam *flowchart* dapat dilihat pada Table 2.1.

Tabel 2. 1 Simbol *Flowchart*

No	Simbol	Keterangan
1	 ***Flow Direction Symbol***	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga *connecting line*
2	 ***Terminator Symbol***	Yaitu simbol untuk permulaan (*start*) atau akhir (*end*) dari suatu kegiatan
3	 ***Processing Symbol***	Yaitu simbol yang menunjukan pengolahan yang dilakukan oleh komputer
4	 ***Manual Operation Symbol***	Yaitu simbol yang menunjukan pengolahan data yang tidak dilakukan oleh komputer
		Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada

5	***Decision Symbol***	
6	 ***Input-Output Symbol***	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya
7	 ***Document Symbol***	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas

2.2.6 *Unified Modeling Language* (UML)

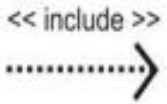
(UML) *Unified Modeling Language* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML *(Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek[12].

a. Use Case

Use Case adalah deskripsi fungsi sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem tersebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem[13]. Simbol *Use Case* dapat dilihat pada Table 2.2

Tabel 2. 2 Simbol *Use Case Diagram*

No.	Simbol	Nama Simbol dan Keterangan

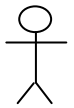
No.	Simbol	Nama Simbol dan Keterangan
1.		*Actor,* menunjukan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
2.		*Include*, relasi tambahan ke sebuah *use case* dimana *use case* ini untuk menjalankan fungsi atau sebagai syarat dijalankannya *use case* ini.
3.		*Association,* simbol yang menghubungkan antara objek satu dengan objek lainnya
4.		*UseCase,* menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case.*

b. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukan sejumlah obyek dan *message* yang diletakan antara obyek-obyek didalam *use case.*

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama *Message* diwakili oleh garis dengan tanda panah[14]. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Table 2.3.

Tabel 2. 3 Simbol *Sequence Diagram*

No.	Simbol	Nama Simbol dan Keterangan
1.		*Actor,* Menggambarkan orang yang sedang berinteraksi dengan sistem.
2.		*Entity Class*, Menggambarkan hubungan kegiatan yang akan

No.	Simbol	Nama Simbol dan Keterangan
		dilakukan.
3.		*Boundary Class*, Menggambarkan sebuah gambaran dari from.
4.		*Control Class*, Menggambarkan penghubung antara boundary dengan tabel.
5.		*A focus of Control & Alife line,* Menggambarkan tempat mulai dan berakhirnya *message.*
6.		*A massage,* menggambarkan pengiriman tugas

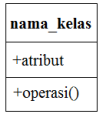
c. Class Diagram

Class Diagram mengambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas dibuat agar *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Class diagram membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* berperan dalam menangkap struktur dari semua kelas

yang membentuk arsitektur sistem yang dibuat [15]. Berikut beberapa simbol dari *class diagram*.

Tabel 2. 4 Simbol *Class Diagram*

No.	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem.
2.		Asosiasi berarah/ *Directed association*	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
3.		Agregasi / *Aggregation*	Relasi antarkelas dengan makna semua-bagian.

2.2.7 Pemrogaman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan projek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek.

Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek – objek dunia nyata. Ada banyak cara untuk memodelkan objek seperti mulai dan abstraksi objek, kelas, hubungan antarkelas sampai abstraksi sistem[16]. Berikut adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek :

a. Kelas (*class)*

Kelas adalah kumpulan objek – objek dengan karakteristik yang sama. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

b. Objek (*object*)

Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal – hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Secara teknis, sebuah objek saat program dieksekusi maka akan dibuat sebuah objek.

c. Metode (*method*)

Operasi atau metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi.Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari *event,* aktivitas atau aksi keadaan, fungsi atau kelakuan dunia nyata. Contoh metode atau operasi misalnya *read*, *write*, *move*, *copy*, dan sebagainya. Kelas sebaiknya memiliki metode *get* dan *set* untuk setiap atribut agar konsep enkapsulasi tetap terjaga. Metode *get* digunakan untuk memberikan akses kelas lain dalam mengakses atribut, dan *set* adalah metode yang digunakan untuk mengisi atribut, agar kelas lain tidak mengakses atribut secara langsung.

d. Atribut (*attribute)*

Atribut dari sebuah kelas adalah variable global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen – elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privasi untuk menjaga konsep enkapsulasi.

e. Pewarisan (*inheritance)*

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya

f. Antarmuka (*interface)*

Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain. Sebuah kelas dapat mengimplementasikan lebih dari satu antarmuka dimana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode - kode program kelas itu.

g. *Reusability*

Pemanfaatan kembali objek yang sudah didefinikasikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

h. Polimorfisme

Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program

2.2.8 Basis Data

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangun data untuk penginformasikan suatu perusahaan instansi, dalam bahasan tertentu.

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupkan fakta mengenai obyek, orang, dan lain-lain[17]. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol)
Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan(redundancy) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar *Structured Query Language* (SQL). MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa kita gunakan adalah MySQL *Free Software* yang berada dibawah Lisensi *General Public License* (GPL).

Dalam penggunaan *SQL* terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam database[18]. Secara garis besar, *SQL Server* mempunyai 3 (Tiga) jenis perintah *SQL* yaitu :

1.) *Data Definition Language* (DDL)

DDL adalah sub perintah dari bahasa SQL yang digunakan untuk membangun kerangka sebuah database, dalam hal ini database dan table. Terdapat tiga perintah penting dalam DDL, yaitu :

a.) CREATE: perintah ini digunakan untuk membuat, termasuk di dalamnya membuat database baru, tabel baru view baru, dan kolom baru. Contoh: CREATE DATABASE nama_database.

b.) ALTER: perintah ALTER berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup di dalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus

kolom, dan memberikan atribut pada kolom. Contoh: ALTER TABLE nama_tabel ADD nama_kolom datatype.

c.) DROP: perintah DROP berfungsi untuk menghapus database atau tabel. Contoh: DROP DATABASE nama_database.

2) ***Data Manipulation Language*** **(DML)**

DML adalah sub perintah dari bahasa SQL yang digunakan untuk memanipulasi data dalam database yang telah dibuat. Terdapat 4 (Empat) perintah penting dalam DML, yaitu :

a.) INSERT: perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika database dan tabel sudah dibuat.

```
INSERT INTO nama_tabel VALUES (data1, data2, dst…);
```

b.) SELECT: perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi.

```
SELECT nama_kolom1, nama_kolom2 FROM nama_tabel;
```

c.) UPDATE: perintah *update* digunakan untuk memperbarui data pada sebuah tabel.

```
UPDATE nama_tabel SET kolom1=data1, kolom2=data2,…
WHERE kolom=data;
```

d.) DELETE: perintah delete digunakan untuk menghapus data dari sebuah tabel.

```
DELETE FROM nama_tabel WHERE kolom=data;
```