BAB II

TINJAUAN PUSTAKA

Beberapa tinjauan pustaka yang digunakan dalam laporan ini adalah sebagai berikut:

1. Sistem Informasi
2. Aplikasi *Desktop*
3. *Database*
4. *Context Diagram*
5. *Data Flow Diagram* (DFD)
6. *Entity Realitionship Diagram* (ERD)
7. *Visual Basic*
8. SQL Server

Berikut merupakan penjabaran tiap tinjauan pustaka yang telah disebutkan sebelumnya:

2.1 Sistem Informasi

Menurut Jogianto (2005: 2) mengemukakan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi. Jadi sistem dapat dikatakan sebagai kumpulan komponen yang memiliki alur informasi untuk mencapai sebuah tujuan yang telah ditetapkan.

Sedangkan sistem informasi adalah kumpulan dari subsistem apapun baik fisik ataupun nonfisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang memiliki arti dan berguna. Menurut Jeperson (2015) sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang

dibutuhkan. Sistem informasi terdiri dari informasi tentang orang-orang, tempat, dan hal penting dalam organisasi atau lingkungan yang melingkupinya. Komponen-komponen sistem informasi adalah:

1. Hardware (perangkat keras).
2. Software (perangkat lunak).
3. Prosedur yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data untuk menghasilkan output.

2.2 Aplikasi Desktop

Desktop Application atau Aplikasi Berbasis Desktop merupakan suatu aplikasi atau *software* milik desktop (PC dan laptop) yang mampu beroperasi tanpa terhubung dengan koneksi internet (*offline*). Untuk menggunakannya, *user* harus menginstalnya terlebih dahulu di sistem operasi pada laptop maupun komputer. Aplikasi berbasis desktop dibuat dengan menggunakan 3 bahasa pemrograman yaitu .Net, Java, dan Delphi. Dimana bahasa pemorgraman .Net meliputi Visual Basic (VB), C++, dan C#.

Pengertian aplikasi desktop juga diungkapkan oleh Dew Omen (2013) yaitu *desktop application* atau aplikasi desktop adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi internet disuatu komputer otonom. Maka dari itu, penulis dapat menyipulkan bahwa aplikasi desktop adalah sistem yang dapat berjalan tanpa membutuhkan jaringan internet namun dengan syarat komputer tersebut telah memasang aplikasi tersebut.

2.3 *Database*

Basis data atau *database* merupakan kegiatan sistem program komputer untuk berbagai aplikasi komputer. Dalam basis data dibutukan suatu media simpan komputer yang terorganisir sedemikian rupa dan juga pemeliharaan data baik dalam fungsi manajemen sistem. Pandangan lain bahwa basis data adalah suatu pengetahuan tentang organisasi data, sehingga database merupakan salah satu

komponen yang penting dalam sistem informasi. Penerapan database dalam sistem informasi disebut dengan sistem basis data (*database system*).

Setiap basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik. Misalnya ada basis data akademik, kepegawaian, inventori dan lain-lain. Sementara dalam basis data akademik kita dapat menempatkan *file* mahasiswa, mata kuliah, dosen kehadiran, nilai dan lain-lain. Karena itu operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat berupa:

1. Pembuatan basis data baru (*create database*)
2. Penghapusan basis data (*drop database*)
3. Pembuatan *file*/tabel baru ke suatu basis data (*create table*)
4. Penghapusan *file*/tabel dari suatu basis data (*drop table*)
5. Penambahan/pengisian data baru ke sebuah *file*/tabel (*insert*)
6. Pengambilan datda dari sebuah *file*/tabel (retrieve/search)
7. Pengubahan data dari sebuah *file*/tabel (*update*)
8. Penghapusan data dari sebuah *file*/tabel (delete)

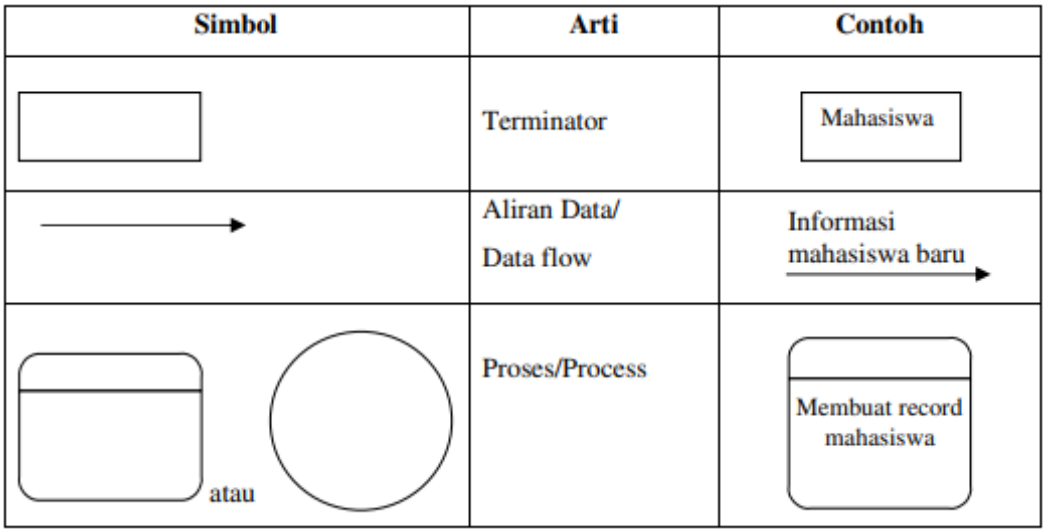
2.4 *Context Diagram*

Menurut Anggraini(2017), *Context diagram* merupakan tingkatan tertinggi dalam diagram aliran data dengan memuat satu proses dan merepresentasikan sistem secara keseluruhan. *Context diagram* ini harus berupa suatu pandangan, yang mencakup masukan-masukan dasar, sistem-sistem dan keluaran. Satu proses tersebut akan diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data-aliran data utama menuju dan dari sistem.

Context diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-aliran data menuju dan dari sistem diketahui melalui penganalisisan dari wawancara dengan user dan sebagai hasil analisis dokumen. *Context diagram* yang dimulai dari penggambaran terminator, aliran data, aliran control penyimpanan, dan proses tunggal menunjukkn keseluruhan sistem.

Terminator ditunjukkan dalam bentuk persegi panjang dan berkomunikasi langsung dengan sistem melalui aliran data atau penyimpanan eksternal antar

terminator tidak diperbolehkan komunikasi langsung. Realitanya hubungan antar terminator dilakukan, tetapi secara definitif karena terminator adalah bagian dari lingkungan, maka tidak relevan jika dibahas dalam *context diagram*.



Gambar 2.1 Simbol Context Diagram

Dalam *context diagram* terdapat beberapa aturan seperti pada poin pertama, jika terdapat banyak terminator yang mempunyai banyak masukan dan keluaran diperbolehkan untuk digambarkan lebih dari satu kali sehingga mencegah penggambaran yang terlalu rumit, dengan ditandai secara khusus untuk menjelaskan bahwa terminator yang dimaksud adalah identik. Tanda tersebut dapat berupa asterik (*) atau pagar (#).

Pada point kedua, jika terminator mewakili individu sebaiknya diwakili oleh peran yang dimainkan personil tersebut. Alasan pertama adalah personil yang berfungsi untuk melakukan itu dapat berganti sedang *Context diagram* harus tetap akurat walaupun personil berganti. Alasan kedua adalah seorang personil dapat memainkan lebih dari satu peran.

Selain itu pada point terakhir menyatakan bahwa adanya focus utama adalah mengembangkan model, maka penting untuk membedakan sumber *(resource*) dan pelaku *(handler)*. Pelaku adalah mekanisme, perangkat atau media fisik yang

mentransportasikan data ke/dari sistem, karena pelaku seringkali familier dengan pemakai dalam implementasi sistem berjalan, maka sering menonjol sebagai sesuatu yang harus digambarkan lebih dari sumber data itu sendiri. Sedangkan sistem baru dengan konsep pengembangan teknologinya membuat pelaku menjadi sesuatu yang tidak perlu digambarkan.

Aliran dalam *context diagram* yaitu memodelkan masukan ke sistem dan keluaran dari sistem seperti halnya sinyal kontrol yang diterima atau dibuat sistem. Aliran data direpresentasikan hanya untuk pendeteksi kejadian dalam lingkungan dengan sistem yang memberkan respon atau membutuhkan data untuk menghasilkan respon. Selain itu dapat digunakan sebagai gambaran transportasi antara sistem dan terminator. Oleh sebab itu, aliran data direpresentasikan jika data tersebut diperlukan untuk menghasilkan respon pada kejadian tertentu sehingga ini seharusnya penggambaran *context diagram* harus dengan asumsi bahwa masukan disebabkan dan diawali oleh terminator, sedangkan keluaran disebabkan dan diawali oleh sistem.

2.5 *Data Flow Diagram* (DFD)

Data Flow Diagram atau DFD ialah diagram aliran data yang digunakan sebagai alat menggambarkan aliran data melalui sistem dan atau pengolahan yang dilakukan oleh sistem. DFD juga dapat dikatakan sebagai penggambaran grafis atas sumber dan tujuan data, yang dapat memperlihatkan data berasal dari mana dan menuju ke mana. *Data Flow Diagram* digunakan untuk mendokumentasikan sistem yang ada serta digunakan merencanakan rancangan dan desain sistem yang baru

Data Flow Diagram dimulai dari diagram konteks yaitu diagram yang berisi gambaran umum dari sistem. DFD dipopulerkan oleh DeMacro & Yordan (1979) dan Gane & Sarson (1979) dengan menggunakan pendekatan Metoda Analisis Sistem Terstruktur. DFD ini merupakan model proses. Model proses merupakan

teknik untuk mengorganisasikan dan mendokumentasikan struktur dan alir data di dalam sistem.

Dalam merepresentasikan diagram aliran data dapat menggunakan simbol yang terdapat pada gambar dibawah. Pada simbol *Data Flow Diagram* (DFD) terdapat proses *(process)*. Proses adalah kegiatan atau fungsi bisnis di mana manipulasi dan transformasi data terjadi. Suatu proses dapat didekomposisi ke tingkat rincian yang lebih halus, untuk mewakili bagaimana data sedang diproses dalam proses. Kedua terdapat penyimpanan data *(data store)* yang artinya penyimpanan data persisten yang diperlukan dan / atau diproduksi oleh proses. Berikut adalah beberapa contoh penyimpanan data: formulir keanggotaan, tabel database, dll.

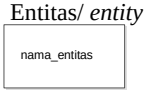
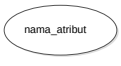
Pada simbol ketiga terdapat entitas eksternal *(external entity)* dapat mewakili manusia, sistem atau subsistem. Di sinilah data tertentu berasal atau pergi ke. Ini adalah eksternal dari sistem yang kita pelajari, dalam hal proses bisnis. Untuk alasan ini, orang biasa menggambar entitas eksternal di tepi diagram. Simbol terakhir yaitu aliran data *(data flow).* Aliran data mewakili aliran informasi, dengan arahnya diwakili oleh panah yang menunjukkan di ujung konektor aliran.

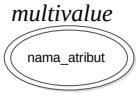
Gambar 2.2 Simbol Data Flow Diagram (DFD)

Dalam pembuata *Data Flow Diagram* (DFD) terdapat beberapa syarat yaitu pemberian nama untuk tiap komponen DFD, pemberian nomor pada komponen proses, menghindari kerumitan dalam pembuatan DFD, memastikan DFD yang dibentuk itu konsisten secara logika. Dengan hal tersebut DFD dapat direpresentasikan dengan tujuan memberikan indikasi mengenai bagaimana data ditransformasi pada saat data bergerak melalui sistem dan menggambarkan fungsi-fungsi (dan sub fungsi) yang mentransformasi aliran data.

2.6 *Entity Realitionship Diagram* (ERD)

Menurut Prijambodo (2018), ERD (*Entity Relation Diagram)* merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data. ERD (*Entity Relation Diagram)* berguna untuk menggambarkan gambaran dari dunia nyata yang akan diterapkan pada suatu *database* sebuah sistem. ERD melihat objek nyata dapat sebagai sebuah entitas – entitas yang memiliki relasi antara entitas yang satu ataupun yang lain. Dengan ERD sendiri dapat membantu mengurangi kesalahan – kesalahan dalam melakukan perancangan *database* dari gambaran dunia nyata dan struktur *database* seperti redudansi data, hubungan – hubungan antara entitas, dan lain sebagainya.

Simbol	*Deskripsi*
Entitas/ *entity* 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh Sistem Informasi komputer. Penamaan entitas biasanya lebih ke kata benda dan belum tentu merupakan nama *table.*
Atribut 	*Field* atau kolom data yang akan disimpan dalam suatu entitas
Atribut kunci primer 	*Field* atau kolom data yang akan disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan, biasanya berupa id. Kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)

Atribut multinilai/ *multivalue* 	*Field* atau kolom data yang akan disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas , biasanya diawali dengan kata kerja
Asosiasi / *association* 	Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antar entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalnya ada kardinalitas 1 ke N atau sering disebut *one to many* menghubungkan entitas A dengan entitas B

Tabel 2.1 Simbol ERD

2.7 *Visual Basic*

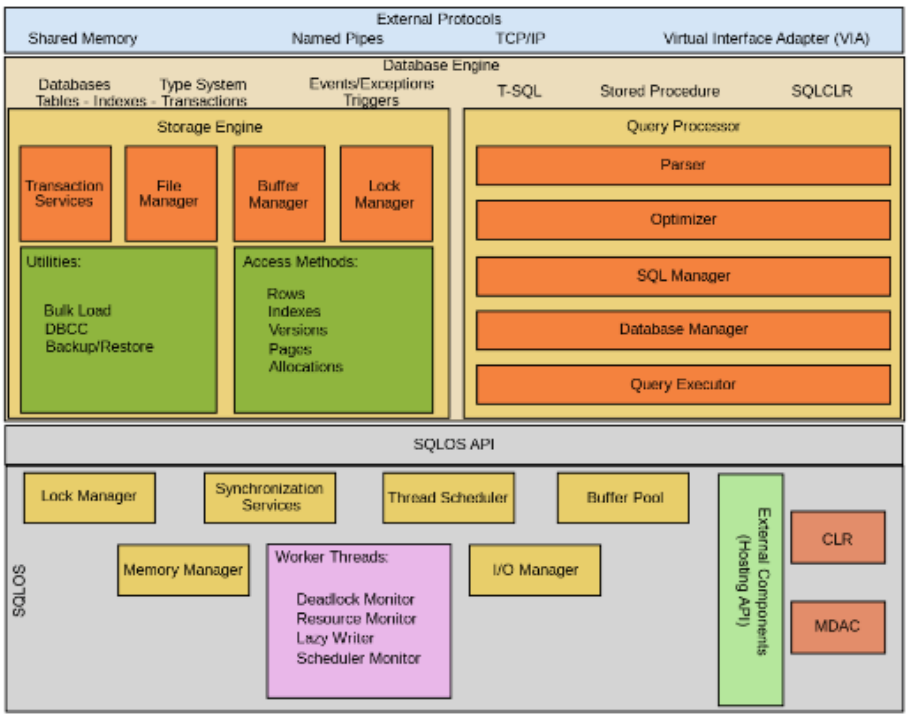
Dilansir dari Techopedia (2018), *Visual Basic* adalah sebuah bahasa pemrograman yang dibuat dan dikembangkan oleh *Microsoft* dengan format sederhana dan mudah dipahami. Inilah mengapa *programmer* pemula menganggap *Visual Basic* sebagai titik awal pengembangan *software*.

Visual Basic atau VB mencakup berbagai alat visual yang bisa dipakai untuk membuat aplikasi tingkat lanjut. GUI (*Graphical User Interface*) dalam *Visual Basic* juga diperluas sehingga VB tidak sekadar bahasa pemrograman. Melainkan mencakup berbagai *library* yang bermanfaat untuk membuat program berorientasi objek. Program ini melibatkan tim pengembangan besar yang bisa mengerjakan proyek bersamaan.

Kegunaan Visual Basic adalah untuk membuat program berbasis *Windows* mulai yang sederhana sampai pemrograman yang lebih kompleks. Contohnya adalah pembuatan aplikasi kasir atau perpustakaan. Untuk membuat aplikasi sederhana dengan visual basic maka kita harus menguasai bahasa pemrograman C++. *Visual Basic* yang paling banyak digunakan adalah *Microsoft Visual Basic*.

2.8 SQL Server

SQL Server adalah sistem manajemen database relasional atau disingkat RDBMS besutan *Microsoft*. Mirip dengan produk pasaran RDBMS lainnya, SQL server ini dibangun di atas SQL yakni, bahasa pemrograman standar untuk berinteraksi dengan database relasional menurut Husni, *et.al*(2022). Merujuk pada sumber DQLab(2022) dalam arstitektur SQL, SQL server terdiri dari dua komponen utama yaitu *database engine*, dan SQLOS. Dimana database engine merupakan komponen inti dari SQL server yang terdiri dari relational engine untuk memproses kueri dan storage engine untuk mengelola file database, halaman indeks, dll. Objek database seperti prosedur tersimpan, rampilan dan pemicu juga di buat dan dijalankan oleh databse engine sedangkan untuk SQLOS terletak dibawah relational engine dan storage engine. SQLOS menyediakan banyak layanan sistem operasi seperti operasi memori, dan manajemen I/O. Layanan lain juga termasuk layanan penganganan dan sinkronisasi pengecualian.



Gambar 2.3 Arsitektur SQL Server DQLab(2022)

Mengacu pada sumber yang sama, SQL Server memiliki kekurangan dan kelebihan dimana pada kelebihannya, SQL Server mendukung banyak sekali *software database* yang menjadi keuntungan bagi penggunanya dalam mempermudah pekerjaan mengolah database. Pengguna dapat menggunakan software database apapun yang ingin digunakan, tanpa perlu khawatir tidak akan kompatibel dengan Microsoft SQL Server. SQL Server juga sering digunakan dalam melakukan clustering data, sehingga data-data yang diperoleh cenderung lebih mudah diimplementasikan. Selain itu kerugian yang dimiliki yaitu SQL server ini merupakan software besutan Microsoft yang artinya harga jualnya memang cukup tinggi jika dibandingkan software lainnya. Selain itu pengguna Mac OS, SQL server tidak bisa digunakan terlebih dahulu yang disebabkan tidak adanya dukungan multi OS.

Secara umum RDBMS menggunakan SQL untuk mengakses database, tapi produk RDBMS tidak hanya satu jenis melainkan berbagai jenis produk, maka dari itu SQL *syntax* pun bisa jadi sedikit berbeda untuk setiap produk tersebut. Berikut contoh perbandingan MySQL, Oracle, dan SQL SERVER untuk menampilkan beberapa baris data dari suatu tabel, seperti pada gambar dibawah bersumber dari DQLab(2022).

MySQL	Oracle	SQL Server
SELECT field(s) FROM table_name LIMIT N;	SELECT field(s) FROM table_name WHERE ROWNUM <= N	SELECT TOP(N) field(s) FROM table_name

Gambar 2.4 Penulisan Sintaks SQL