

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam Penelitian ini, penulis melakukan analisis literatur dari penelitian-penelitian sebelumnya sebagai sumber dukungan untuk penelitian yang akan dilakukan oleh penulis. Berikut ini adalah review literatur yang telah diteliti sebelumnya :

Tabel 2. 1 Daftar Literatur

No	Penulis	Tahun	Judul
1	Rosalia O.B, Siahaan M	2022	Perancangan dan implementasi sistem kasir digital pada UMKM <i>Coffee Lucky Star</i> berbasis web
2	Arafat F, Rizkiansah F, Rosyani P	2021	Penerapan aplikasi kasir berbasis android pada UMKM “Nangkringan” Bintaro Pesanggrahan
3	Mulyani A, Setiawan R, Rusmana R.A	2022	Rancang bangun aplikasi kasir penjualan pada usaha mikro kecil menengah 3Manstore berbasis web
4	Faturohman D, Eosina P, Riana F	2018	Perancangan aplikasi kasir berbasis android (studi kasus Toko Usaha Tani Leuwiliang)
5	Saputra A, dkk	2021	Pengembangan aplikasi kasir berbasis web menggunakan metode <i>Waterfall</i>

2.1.1 Tinjauan Literatur 1

Oleh Rosalia O.B, Siahaan M, (2022), Universitas Internasional Batam. Perancangan dan implementasi sistem kasir digital pada UMKM *Coffee Lucky Star*. *Coffee Lucky Star* merupakan salah satu Usaha Mikro Kecil Menengah (UMKM) di kota Batam yang didirikan oleh Bapak Refendy pada pertengahan tahun 2020 dan berlokasi di Komplek Bumi Indah Nagoya, Batam. Dalam kegiatan akuntannya *Coffee Lucky Star* masih menggunakan sistem pembukuan secara manual. Mulai dari pencatatan pesanan, proses pembayaran dan laporan keuangan harian hingga bulanan semua masih dilakukan secara manual dengan menggunakan buku. *Coffee Lucky Star* ini juga belum menerapkan sistem keuangan terperinci yang disertai pencatatan baik dalam pemasukannya maupun pengeluarannya. Sehingga *Coffee Lucky Star* ini tidak dapat memproyeksi laba rugi secara cepat dan tepat. Penelitian ini bertujuan untuk memberikan solusi yang tepat atas permasalahan yang dihadapi *Coffee Lucky Star* untuk meningkatkan efektivitas dan efisiensi operasional bisnisnya terutama pada sistem keuangannya. Sistem kasir digital berbasis website ini dikembangkan dengan menggunakan bahasa pemrograman web PHP dan database yang digunakan adalah XAMPP. Dengan adanya sistem pencatatan akuntansi berbasis website ini menjadi solusi terbaik untuk mempermudah pencatatan laporan keuangan pada *Coffee Lucky Star*.

2.1.2 Tinjauan Literatur 2

Oleh Arafat F, Rizkiansah F, Rosyani P, (2021), Teknik Informatika, Universitas Pamulang. Penerapan aplikasi kasir berbasis android pada UMKM

“Nangkringan” Bintaro Pesanggrahan. Kasir merupakan salah satu aspek penting dalam hal transaksi yang dilakukan oleh para pihak usaha jual beli. Hal yang dapat menghambat proses transaksi yaitu dikarenakan transaksi tersebut masih menggunakan dengan cara manual yang tidak efektif dan efisien di zaman sekarang yang serba canggih ini. Sering kali cara manual banyak terjadi kesalahan baik itu perhitungan maupun hal lain seperti hilangnya catatan transaksi yang dapat menyebabkan tidak bisa menghitung pendapatan bulanan. Hasil yang dapat dihasilkan oleh aplikasi kasir yang hendak dibuat oleh penulis ini dapat membantu para pihak usaha jual beli dalam hal transaksi agar lebih efektif dan efisien sehingga penjualan dapat meningkat dengan adanya aplikasi kasir tersebut.

2.1.3 Tinjauan Literatur 3

Oleh Mulyani A, Setiawan R, Rusmana R.A, (2022), Jurnal Algoritma, Institut Teknologi Garut. Rancang bangun aplikasi kasir penjualan pada usaha mikro kecil menengah 3Manstore berbasis web. Aplikasi kasir penjualan pada usaha mikro kecil menengah merupakan jawaban terhadap permasalahan yang ada pada salah satu usaha mikro kecil menengah pengrajin tas yang berada di daerah Leles Kabupaten Garut. Hasil dari observasi dan wawancara dengan pemilik usaha mikro kecil menengah permasalahan yang ada adalah proses transaksi masih manual menggunakan nota tulis sehingga sangat mudah untuk nota hilang atau rusak, proses perhitungan laba yang kadang tidak balance dikarenakan human error saat proses rekapitulasi serta tidak adanya catatan hutang sehingga saat proses pembayaran hutang reseller kadang tidak sinkron. Tujuan dibuatnya aplikasi ini adalah untuk membantu proses kegiatan penjualan pada usaha mikro kecil menengah, membantu

mengelola hutang piutang, dan membantu proses rekapitulasi laporan laba rugi. *Rational Unified Process* (RUP) yang di mana tahapan pelaksanaannya hanya meliputi *inception*, *elaboration*, dan *construction* menjadi bagian dari metodologi yang digunakan. Kemudian menggunakan pemodelan *Unified Modeling Language* (UML) dan pengujian alpha dengan menggunakan metode *Black Box Testing* dilakukan pada tahapan *construction*, sedangkan untuk bahasa pemrograman ialah PHP dan MySQL sebagai basis datanya. Hasil dari penelitian ini berupa Aplikasi Kasir Penjualan Untuk Usaha Mikro Kecil Menengah Berbasis Web yang memilih hak akses multi-user serta sudah dilengkapi dengan fitur transaksi, manajemen hutang dan laporan laba rugi.

2.1.4 Tinjauan Literatur 4

Oleh Fatur Rahman D, Eosina P, Riana F, (2018), Teknik Informatika, Universitas Ibn Khaldun. Perancangan aplikasi kasir berbasis android (studi kasus Toko Usaha Tani Leuwiliang). Toko Usaha Tani merupakan sebuah toko pupuk yang berlokasi di pasar lama leuwiliang yang masih menggunakan sistem pelayanan kasir yang sangat sederhana. Hasil transaksi dihitung menggunakan kalkulator dan dicatat pada kwitansi. Dalam perhitungan transaksi masih sering terjadi kesalahan, dan sering hilangnya data-data penjualan yang menyebabkan kesalahan perhitungan modal dan keuntungan. Perkembangan aplikasi di bidang bisnis atau perdagangan yang saat ini menarik untuk digunakan dalam organisasi maupun perusahaan yang dapat mengembangkan usaha dengan baik dengan adanya sistem. Android adalah sistem operasi mobile yang sangat populer di Indonesia, tercatat kurang lebih 41 juta pengguna android di indonesia yang jika dibandingkan dengan pengguna IOS

yang hanya 21 juta pengguna. Aplikasi kasir yang akan di bangun memiliki fungsi untuk menghitung transaksi secara otomatis, menyimpan data penjualan, dan mengeluarkan struk pembayaran dengan bluetooth printer. Aplikasi kasir berbasis android ini diharapkan dapat mempermudah pengguna untuk menghitung hasil transaksi dengan lebih cepat dan efisien dan juga pengguna dapat mengatasi hilangnya data-data penjualan karena dengan aplikasi kasir berbasis android ini data penjualan tersimpan dengan baik didalam database.

2.1.5 Tinjauan Literatur 5

Oleh Saputra A,dkk, (2021), Teknik Informatika, Universitas Pamulang. Pengembangan aplikasi kasir menggunakan metode *Waterfall*. Perusahaan yang bergerak dalam bidang jasa percetakan masih banyak yang menggunakan proses manual dalam transaksinya, proses sistem tersebut kurang efisien dan kemungkinan kesalahan dalam input data masih dapat terjadi. Seperti proses pencatatan pemesanan menggunakan kertas, perhitungan menggunakan kalkulator dan transaksi penjualan ditulis pada buku. Maka itu untuk menyelesaikan permasalahan tersebut dibutuhkan sebuah aplikasi kasir yang dapat memudahkan dalam proses pemesanan, perhitungan dan transaksi secara akurat dengan aplikasi kasir berbasis web yang memiliki fitur login, transaksi penjualan, data produk, tambah produk, pengaturan dan keluar aplikasi. Proses pengembangan kami menggunakan metode *waterfall* dan untuk menggambarkan rancangan sistem kami menggunakan *Unified Modeling Language* (UML). Penerapan dengan menggunakan metode *waterfall* dapat memudahkan dalam menyelesaikan proses pengembangan aplikasi karena sudah adanya tahapan-tahapan pada metode tersebut.

2.2 Teori Umum

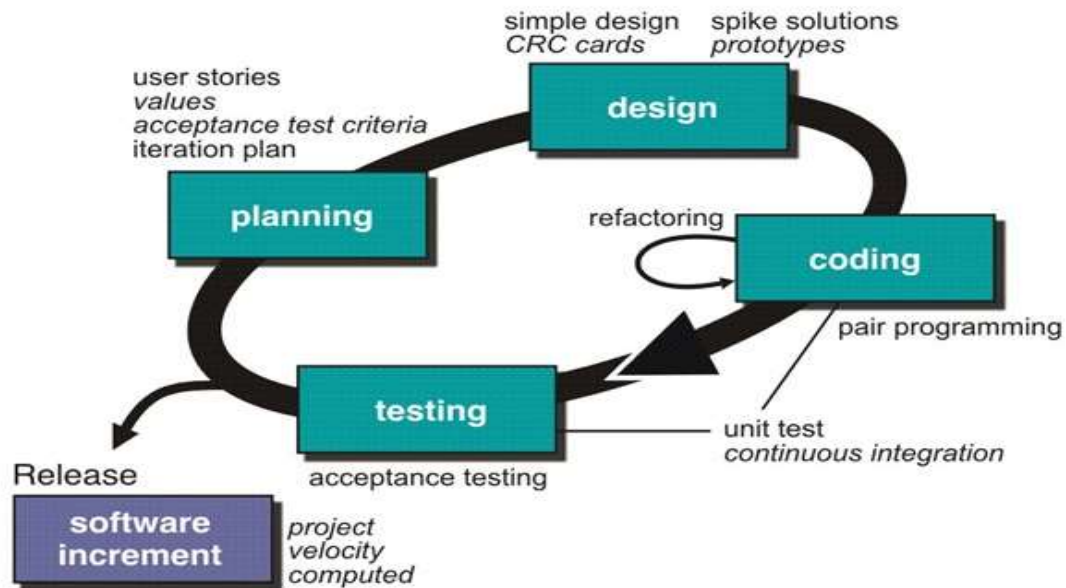
2.2.1. *jQuery Mobile*

jQuery Mobile adalah *framework* yang dioptimalkan untuk perangkat-perangkat layar sentuh (*touch-optimized*). Sebagai pustaka, *jQuery Mobile* juga dikenal sebagai *JavaScript* maupun *framework mobile* yang juga dikenal sebagai *library JavaScript* maupun *Framework mobile* ini dikembangkan oleh tim *jQuery Project*. Tujuan pengembangannya adalah berfokus pada pembuatan *framework* yang kompatibel dengan beraneka ragam perangkat *smartphone* dan tablet. *jQuery Mobile* dikembangkan sebagai alternatif yang menjawab kebutuhan akan pengembangan aplikasi web mobile yang mudah dan memiliki banyak potensi. *jQuery Mobile* sangat mungkin untuk membuat aplikasi web mobile yang penampilan dan interaktivitasnya konsisten di semua perangkat yang mendukung. Serta memiliki kemampuan *user interface* (UI) yang canggih. Kelebihan *jQuery* adalah kodenya ringan, menerapkan desain yang dapat beradaptasi dengan berbagai jenis perangkat mobile, dan menonjolkan markup semantic. Oleh karena itu, desainnya sangat *fleksibel* dan mempunyai *Theme* yang bisa disesuaikan. Hal ini membuat *jQuery Mobile* sangat sekali digunakan (Camden and Matthews, 2017).

2.2.2 Model Pengembangan Sistem

Menurut Pressman (2019) *extreme programming* merupakan suatu pendekatan berorientasi objek dan sebagai pengembangan perangkat lunak cepat sedikit lebih rinci dengan tujuan memberikan ulasan secara ringkas.

Pengembangan *extreme programming* dapat dilihat menggunakan suatu alur tahapan pengembangan yang dapat dilihat pada Gambar 2.1:



Gambar 2. 1 *Extreme Programming*
Sumber (Pressman, 2019)

Berdasarkan tahapan tersebut merupakan suatu paradigma yang diinginkan mencakup didalam seperangkat aturan dan praktik-praktik dalam empat konteks kegiatan kerangka kerja yaitu :

1. Perencanaan

Customer dan *XP team* bekerja bersama untuk memutuskan bagaimana grup *story* untuk *release* berikutnya (*software increment*) untuk dibangun oleh *XP team*.

Jika komitmen telah dibuat, *XP team* akan membangun *story* dengan cara :

Semua *story* segera diimplementasikan (dalam beberapa minggu)

- a. *Story* dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama.
- b. *Story* dengan resiko paling tinggi akan diimplementasikan terlebih dulu. Setelah *project* pertama di *release* dan di *delivery*, *XP team* memperhitungkan kecepatan *project*. Selama *development*, *customer*, dapat menambah *story*, merubah *value*, membagi *story* atau menghapusnya.

2. Perancangan

Penggunaan rancangan bertujuan untuk membangun dan menggambarkan sistem yang akan dibuat guna mempermudah proses pengkodean. *XP* juga menggunakan *CRC card*, untuk mengenali dan mengatur *object oriented class* yang sesuai dengan *software increment*.

3. Pengkodean

Sebelum membuat *code*, lebih baik membuat unit *test* tiap *story* untuk dimasukkan dalam *software increment*. *XP* menyarankan agar dua orang bekerja bersama pada satu *computer workstation* untuk membuat *code* dari satu *story* (*pair programming*), untuk menyediakan *real time problem solving* dan jaminan *real time quality*. Setelah *pair programming* selesai, *code* diintegrasikan dengan kerja lainnya (*continuous integration*).

4. Pengujian

Unit *test* yang telah dibuat harus diimplementasikan menggunakan suatu *framework* dan diatur ke dalam *universal testing suite*, *integrasi*, dan *validasi* sistem dapat dilakukan setiap hari. *Customer test (acceptance test)* dilakukan oleh

customer dan fokus pada keseluruhan fitur dan fungsional sistem. *Acceptance test* diperoleh dari *customer stories* yang telah diimplementasikan sebagai bagian dari *software release*.

Kelebihan Model *Extreme Programming*:

Komunikasi dalam XP dibangun dengan melakukan pemrograman berpasangan (*pair programming*). *Developer* didampingi oleh pihak klien dalam melakukan *coding* dan *unit testing* sehingga klien bisa terlibat langsung dalam pemrograman sambil berkomunikasi dengan *developer*. Selain itu perkiraan beban tugas juga diperhitungkan.

1. Menekankan pada kesederhanaan dalam pengkodean: “*What is the simplest thing that could possibly work?*” Lebih baik melakukan hal yang sederhana dan mengembangkannya besok jika diperlukan.
2. Setiap *feedback* ditanggapi dengan melakukan *test*, *unit test* atau *system integration* dan jangan menunda karena biaya akan membengkak (uang, tenaga, waktu).
3. Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki.

Kelemahan Model *Extreme Programming* :

Pengembangan *extreme programming* yang digunakan tentunya masih memiliki beberapa kelemahan atau kekurangan seperti berikut.

1. *Developer* harus selalu siap dengan perubahan karena perubahan akan selalu

diterima.

2. Tidak bisa membuat kode yang detail di awal *prinsip simplicity* dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga.

2.2.3 My SQL

Menurut Nugroho (2018), merupakan *database multiuser* yang menggunakan bahasa *Structure Query Language (SQL)*. *MySQL* dalam operasi *client-server*, melibatkan *server demon MySQL* di sisi *server* dan berbagai macam program, serta *library* yang berjalan disisi *client*. *MySQL* mampu menangani data yang cukup besar. *SQL* adalah bahasa standar yang digunakan untuk mengakses *server database*. Bahasa ini awalnya dikembangkan IBM, tetapi telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan *SQL*, proses akses *database* menjadi lebih *friendly*. Dalam konteks bahasa *SQL*, informasi umumnya tersimpan dalam tabel-tabel yang secara logis merupakan struktur dua dimensi yang terdiri atas baris-baris data yang berada dalam satu atau lebih kolom. Baris pada tabel sering disebut entitas dari kata, sedangkan kolom sering disebut atribut atau field. Keseluruhan tabel itu dihimpun dalam satu kesatuan yang disebut *database*.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL (Structure Query Language)* atau *DBMS* yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi *GNU General Public License (GPL)*,

tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL Tidak sama dengan proyek-proyek seperti Apache, di mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, di mana memegang 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 menggunakan *SQL* sebagai bahasa dasar untuk mengakses *database* yang terbagi menjadi tiga bagian, yaitu DDL, DML, dan DCL.

1. DDL (*Data Definition Language*)

Data Definition Language (DDL) adalah suatu tata Bahasa definisi data pada *MySQL*, DDL digunakan untuk mendefinisikan suatu *database*, *table*, *tablespace*, *log file group*, *server*, dan *index*. DDL umumnya digunakan mendefinisikan suatu wadah data atau *record*. DDL terdiri dari *create*, *alter*, *drop*, *rename* (Sibero, 2013:98).

Berikut penjelasan mengenai masing-masingnya:

a. *Create*

Create digunakan untuk mendefinisikan suatu struktur, wadah yang digunakan sebagai media suatu data/*record* dan atribut pendukung lainnya.

b. *Alter*

Alter digunakan untuk mengubah definisi suatu struktur, wadah yang

digunakan sebagai media suatu data/*record* dan atribut pendukung lainnya.

c. *Drop*

Drop digunakan untuk menghapus suatu struktur, wadah yang digunakan sebagai media suatu data/*record* dan atribut pendukung lainnya.

d. *Rename*

Rename digunakan untuk mengganti nama suatu struktur, wadah yang digunakan sebagai media suatu data/*record*. Penggunaan *rename* hanya berlaku untuk *database* dan *table*.

2. DML (*Data Manipulation Language*)

Data Manipulation Language adalah suatu tata bahasa memanipulasi data pada *MySQL*, DML digunakan untuk memanipulasi data pada *table database* Berikut komponen DML:

a. *Select*

Select berfungsi untuk menyeleksi, memilih atau menampilkan data-data yang ada dalam tabel, baik menampilkan semua kolom maupun sebagian kolom berdasarkan kondisi.

b. *Insert*

Insert berfungsi untuk memasukkan atau menyimpan data dari luar ke sistem dalam *database*. Ada beberapa cara dalam memasukkan data yaitu dengan menyamakan kolom dan data, menyebutkan kolom, tanpa menyebutkan kolom, memasukkan hanya sebagian pada kolom.

c. *Update*

Update berfungsi untuk memperbaharui data lama menjadi data yang baru.

d. *Delete*

Delete berfungsi untuk menghapus atau menghilangkan baris data (*record*) dari *table*.

3. DCL (*Data Control Language*)

Data Control Language adalah bagian inti dari *SQL* yang mempunyai kemampuan untuk mengatur hak akses terhadap sebuah basis data (*database*).

DCL terbagi menjadi 2:

a. *GRANT* (Memberikan Hak Akses)

GRANT merupakan perintah untuk memberikan hak izin akses bagi *user* di *MySQL* agar dapat mengakses *database*, *table* dan *colom*.

b. *REVOKE* (Mencabut Hak Akses)

REVOKE merupakan kebalikan dari perintah *GRANT* yaitu menghapus atau mencabut kembali izin akses *user MySQL* yang sebelumnya telah diberikan. Didalam *SQL*, *Key* terbagi menjadi beberapa jenis diantaranya adalah sebagaiberikut:

1. *Primary Key*

Kunci utama adalah atribut merupakan kunci calon yang telah dipilih untuk mengidentifikasi setiap *record* secara unik. Kunci utama harus merupakan atribut yang benar-benar unik dan tidak boleh ada nilai *NULL*. Kunci utama adalah suatu nilai dalam basis data yang digunakan untuk mengidentifikasi suatu baris dalam *table*.

2. *Foreign Key*

Jika sebuah kunci utama terhubung ke tabel lain, maka keberadaan kunci utama tersebut disebut sebagai kunci tamu. Kunci tamu adalah Sebuah kumpulan atribut dalam satu relasi yang digunakan untuk menunjuk ke suatu baris pada relasi yang lain (harus berkorespondensi dengan kunci utama pada relasi yang kedua), seperti logical pointer.

3. *Candidate Key*

Kunci calon adalah salah satu rangkaian yang mempunyai nilai unik untuk membedakan atau mengidentifikasi nilai-nilai kombinasi yang unik diantara semua kejadian yang spesifik dari entetitas. Kunci calon ini tidak boleh berisi atribut dari tabel yang lain. kombinasi dari atribut yang dapat digunakan untuk mengidentifikasi secara unik rekor database tanpa data apapun yang asing.

4. *Secondary Key*

Kunci sekunder adalah sebuah atribut atau kombinasi yang digunakan hanya untuk tujuan pengambilan data.

5. *Alternate Key*

Kunci Alternatif adalah kunci alternatif yang tidak terpilih. Misal : dalam suatu entitas terdapat dua atribut yang bisa dijadikan sebagai kunci. Sementara yang boleh dijadikan kunci hanya satu, maka anda harus memilih salah satu. Atribut yang dipilih disebut kunci utama. sedangkan atribut yang tidak dipilih disebut dengan kunci.

6. *Composite Key*

Dalam desain basis data, kunci komposit adalah kunci yang terdiri dari 2 atau lebih atribut yang secara unik mengidentifikasi suatu kejadian entitas. Setiap atribut yang membentuk kunci senyawa adalah kunci sederhana dalam haknya sendiri.

Secara garis besar, *database MySQL* mempunyai 3 macam tipe data, yaitu:

1. Tipe Data *Numeric*

Tipe Data *Numeric* pada *database MySQL* terbagi atas beberapa macam tipe data, yaitu:

a. *INT*

INT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan -2.147.483.648 s/d 2.147.483.647. Tipe data ini mempunyai ukuran 4 byte (32 bit).

b. *TINYINT*

TINYINT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -128 s/d 127. Tipe data ini mempunyai ukuran 1 byte (8 bit).

c. *SMALLINT*

SMALLINT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -32.768 s/d 32.767. Tipe data ini mempunyai ukuran 2 byte (16 bit).

d. *MEDIUMINT*

MEDIUMINT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -8.388.608 s/d 8.388.607. Tipe data ini mempunyai ukuran 3 byte (24 bit).

e. *BIGINT*

BIGINT digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -8.388.608 s/d 8.388.607. Tipe data ini mempunyai ukuran 8 byte (64 bit).

f. *FLOAT*

FLOAT digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif persisi tunggal. Tipe data ini mempunyai ukuran 4 byte (32 bit).

g. *DOUBLE*

DOUBLE digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif persisi ganda. Tipe data ini mempunyai ukuran 8 byte (64 bit).

h. *DECIMAL*

DECIMAL digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif persisi ganda. Tipe data ini mempunyai ukuran 8 byte (64 bit).

i. *REAL*

REAL digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif. Tipe data ini mempunyai ukuran 8 byte

(64 bit).

j. *NUMERIC*

NUMERIC digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif. Tipe data ini mempunyai ukuran 8 byte (64 bit).

2. Tipe Data *Date & Time*

Tipe Data *Date & Time* pada *database MySQL* terbagi atas beberapa macam tipe data, yaitu:

a. *DATE*

DATE digunakan untuk menyimpan data tanggal dalam format YY:MM:DD.

b. *DATETIME*

DATETIME digunakan untuk menyimpan data tanggal dan waktu dalam format YY:MM:DD HH:MM:SS.

c. *TIME*

TIME digunakan untuk menyimpan data waktu dalam format HH:MM:SS.

d. *YEAR*

YEAR digunakan untuk menyimpan data tahun.

3. Tipe Data *String*

Tipe Data *String* pada *database MySQL* terbagi atas beberapa macam tipedata, yaitu:

a. *CHAR*

CHAR digunakan untuk menyimpan data karakter/*string* dengan ukuran tetap. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 255 karakter.

b. *VARCHAR*

VARCHAR digunakan untuk menyimpan data karakter/*string* dengan ukuran dinamis. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 255 karakter untuk *MySQL* Versi 4.1 dan mempunyai jangkauan antara 0 s/d 65.535 untuk *MySQL* versi 5.0.3

c. *BLOB*

BLOB adalah tipe data yang digunakan untuk menyimpan data biner seperti *files*, *image*, suara, dll. Tipe data ini mempunyai 216-1 byte.

d. *TINYBLOB*

TINYBLOB digunakan untuk menyimpan data biner seperti *file*, *image*, dan suara. Tipe data ini mempunyai 255 byte.

e. *MEDIUMBLOB*

MEDIUMBLOB digunakan untuk menyimpan data biner seperti *file*, *image*, dan suara. Tipe data ini mempunyai 244-1 byte.

f. *LOB*

LOB digunakan untuk menyimpan data biner seperti *file*, *image*, dan suara. Tipe data ini mempunyai 232-1 byte.

g. *TEXT*

TEXT digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 65.535 (216-1) karakter.

h. *TINYTEXT*

TINYTEXT digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 255 untuk *MySQL* Versi 4.0 dan mempunyai jangkauan antara 0 s/d 65.535 untuk *MySQL* versi 5.0.3.

i. *MEDIUMTEXT*

MEDIUMTEXT digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 224-1 karakter.

j. *LONGTEXT*

LONGTEXT digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 232-1 karakter.

k. *ENUM*

ENUM digunakan untuk menyimpan data *enumerasi* (kumpulan data).

l. *SET*

SET digunakan untuk menyimpan data himpunan data.

Dari pengertian diatas dapat disimpulkan bahwa *MySQL* merupakan aplikasi *database* yang bertujuan memudahkan penyimpanan dan pengaksesan data dan dapat diakses dengan cara yang mudah dan cepat.

2.3 Alat Pengembangan Sistem

Alat pengembangan sistem yang digunakan menggunakan pemodelan *Unified Modeling Language* (UML) *use case diagram* dan *activity diagram*.

2.3.1. Unified Modeling Language (UML)

Unified Modeling Language (UML) 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. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahuddin, 2018).


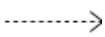

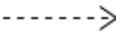
Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisa dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek (Rosa A.S dan M. Shalahuddin, 2018). UML terdiri dari 13 macam diagram yang dikelompokan dalam tiga kategori. Pembagian kategori dan macam-macam *diagram* tersebut dapat dilihat pada gambar dibawah ini.






2.3.2. Use Case Diagram

Use Case Diagram merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Use Case Diagram* dapat

digunakan untuk kebutuhan apa saja yang diperlukan dalam suatu sistem, sehingga sistem dapat digambarkan dengan jelas bagaimana proses dari sistem tersebut, bagaimana cara aktor menggunakan sistem, serta apa saja yang dapat dilakukan pada suatu sistem. Simbol - simbol pada *use case diagram* dapat dilihat pada Tabel 2.2 dibawah ini.

Tabel 2. 2 Simbol *Use Case Diagram*

Simbol	Fungsi	Keterangan
	<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan



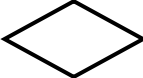

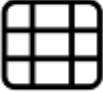
		fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip inheritance pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan.
	<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.
	<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.



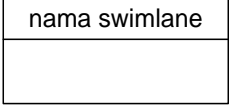
Sumber (Rosa A.S dan M. Shalahuddin, 2018)

2.3.3. Activity Diagram

Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah *activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan actor (Rosa A.S dan M. Shalahuddin, 2018) Simbol-simbol *activity diagram* dapat dilihat pada Tabel 2.3 dibawah ini.

Tabel 2. 3 Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Tabel 	Suatu file komputer dari mana data bisa di baca atau direkam selama kejadian.


<p><i>Dokumen</i></p> 	<p>Menunjukkan dokumen sumber atau laporan.</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

Sumber (Rosa A.S dan M. Shalahuddin, 2018)

2.3.4. *Class Diagram*

Class diagram menunjukkan set kelas, *interface*, kolaborasi dan hubungan mereka. Diagram ini adalah diagram yang paling umum ditemukan dalam sistem pemodelan berorientasi objek, *class diagram* menangani tampilan desain statis dari suatu 10 sistem. *Class diagram* biasanya digunakan untuk menggambarkan perbedaan yang mendasar antara class-class, hubungan antara class, dan di mana sub-sistem class tersebut. Simbol-simbol *class diagram* dapat dilihat pada Tabel 2.4 dibawah ini.

Tabel 2. 4 Simbol *Class Diagram*

Simbol	Fungsi	Keterangan
	<p><i>Generalization</i></p>	<p>Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di</p>

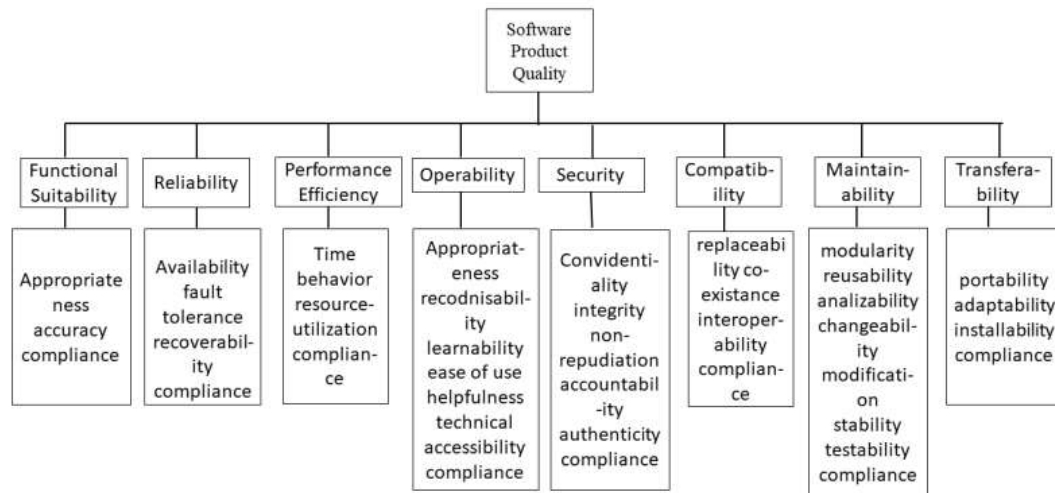
		atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor..
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Sumber (Rosa A.S dan Shalahuddin, 2018)

2.4 Pengujian Sistem ISO 25010

Pada Penelitian ini, metode yang digunakan untuk menguji sistem ini adalah ISO 25010. Standar ISO /IEC 25010 pertama kali diperkenalkan pada tahun 1991 melalui pertanyaan tentang definisi kualitas perangkat lunak. ISO/IEC 25010 memperkenalkan tipe kualitas (*quality in use*) dimana mengikuti elemen yang telah

diketahui. ISO/IEC 25010 merupakan standar internasional yang diterbitkan oleh ISO untuk evaluasi kualitas perangkat lunak dan merupakan perkembangan dari ISO 9126. Model kualitas ISO 25010 mempunyai delapan ukuran kualitas yang ditetapkan oleh ISO/IEC 25010 yang dapat dilihat pada Gambar 2.2



Gambar 2. 2 Karakteristik ISO 25010
Sumber (Kurniawan, Arifianto and Muharom, 2018)

1. *Functional Suitability*

Functional Suitability merupakan tingkat dimana produk perangkat lunak menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika perangkat lunak digunakan dalam kondisi tertentu. Sub karakteristik *Functional Suitability* meliputi *appropriateness*, *accuracy*, dan *compliance*. Berikut ini penjelasan untuk masing-masing sub karakteristik *Functional Suitability* yang dapat dilihat pada tabel 2.5.

Tabel 2. 5 Penjelasan Sub-karakteristik *Functional Suitability*

Karakteristik	Sub karakteristik	Penjelasan
<i>Functional Suitability</i>	<i>Appropriateness</i>	Sejauh mana produk perangkat lunak menyediakan fungsi yang tepat untuk tugas-tugas tertentu dan tujuan pengguna?
	<i>Accuracy</i>	Sejauh mana produk perangkat lunak memberikan hasil yang tepat atau spesifik dengan tingkat presisi yang diperlukan?
	<i>Compliance</i>	Tingkat dimana perangkat lunak mematuhi standar, konvensi, atau peraturan dalam undang-undang dan peraturan serupa yang berkaitan dengan kesesuaian fungsional?

Sumber (Kurniawan, Arifianto and Muharom, 2018)

2. Reliability

Reliability didefinisikan sejauh mana perangkat lunak dapat mempertahankan tingkat kinerja dalam kondisi tertentu. Subkarakteristik *Reliability* meliputi *availability*, *fault tolerance*, *recoverability*, *reliability compliance*. Berikut ini penjelasan untuk masing-masing sub karakteristik *Reliability* yang dapat dilihat pada tabel 2.6

Tabel 2. 6 Penjelasan Sub-karakteristik *Reliability*

Karakteristik	Sub karakteristik	Penjelasan
<i>Reliability</i>	<i>Availability</i>	Sejauh mana komponen perangkat lunak beroperasi dan tersedia saat diperlukan untuk digunakan.
	<i>Fault tolerance</i>	Sejauh mana produk perangkat lunak dapat mempertahankan tingkat kinerja tertentu dalam kasus kesalahan perangkat lunak atau pelanggaran antarmuka yang ditentukan.
	<i>Recoverability</i>	Tingkat di mana produk perangkat lunak dapat menetapkan kembali tingkat kinerja ditentukan dan memulihkan data secara langsung terpengaruh kasus kegagalan.
	<i>Reliability compliance</i>	Sejauh mana produk perangkat lunak mematuhi standar, konvensi atau peraturan yang berkaitan dengan kehandalan.

Sumber (Kurniawan, Arifianto and Muharom, 2018)

3. *Performance efficiency*

Performance efficiency merupakan sejauh mana perangkat lunak memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. Sub karakteristik *Performance efficiency* meliputi *time behaviour*,

resource utilization, performance efficiency compliance. Berikut ini penjelasan untuk masing-masing sub karakteristik *Performance efficiency* yang dapat dilihat pada tabel 2.7.

Tabel 2. 7 Penjelasan Sub-karakteristik *Performance efficiency*

Karakteristik	Sub karakteristik	Penjelasan
<i>Performance efficiency</i>	<i>Time behaviour</i>	Sejauh mana perangkat lunak memberikan respons yang tepat, waktu pemrosesan dan laju keluaran ketika menjalankan fungsinya.
	<i>Resource utilization</i>	Sejauh mana perangkat lunak menggunakan jumlah dan jenis sumber daya yang tepat ketika perangkat lunak menjalankan fungsinya.
	<i>Performance Efficiency Compliance</i>	Sejauh mana perangkat lunak mematuhi standar atau konvensi yang berkaitan dengan efisiensi kinerja.

Sumber (Kurniawan, Arifianto and Muharom, 2018)

4. Operability

Operability merupakan sejauh mana produk perangkat lunak dapat dipahami, dipelajari, digunakan dan menarik bagi pengguna bila digunakan dalam kondisi tertentu. Sub karakteristik *Operability* meliputi *appropriateness recognisability, learnability, ease of use, helpfulness, attractiveness, technical accessibility*, dan

compliance. Berikut ini penjelasan untuk masing-masing sub karakteristik *Operability* yang dapat dilihat pada tabel 2.8.

Tabel 2. 8 Penjelasan Sub-karakteristik *Operability*

Karakteristik	Sub karakteristik	Penjelasan
<i>Operability</i>	<i>Apropriateness</i> <i>Recognisability</i>	Sejauh mana perangkat lunak memungkinkan pengguna untuk mengenali apakah perangkat lunak sesuai dengan kebutuhan pengguna?
	<i>Learnability</i>	Sejauh mana perangkat lunak pengguna untuk mempelajari aplikasinya?
	<i>Ease of use</i>	Sejauh mana perangkat lunak memudahkan pengguna mengoperasikan dan mengendalikan?
	<i>Helpfulness</i>	Sejauh mana perangkat lunak dapat membantu pengguna?
	<i>Attractiveness</i>	Apakah antarmuka terlihat baik?
	<i>Technical Accessibility</i>	Tingkat pengoperasian perangkat lunak untuk pengguna dengan cacat tertentu?
	<i>Compliance</i>	Apakah perangkat lunak sudah mematuhi standar, konvensi, panduan gaya atau peraturan yang berkaitan dengan pengoperasian?

Sumber (Kurniawan, Arifianto and Muharom, 2018)

5. Security

Security merupakan perlindungan item sistem dari akses yang tidak disengaja atau berbahaya, penggunaan, modifikasi, perusakan dan pengungkapan. Subkarakteristik *Security* meliputi *confidentiality*, *integrity*, *non-repudiation*, *accountability*, *authenticity*, *security compliance*. Berikut ini penjelasan untuk masing-masing sub karakteristik *security* yang dapat dilihat pada tabel 2.9.

Tabel 2. 9 Penjelasan Sub-karakteristik *Security*

Karakteristik	Sub karakteristik	Penjelasan
<i>Security</i>	<i>Confidentiality</i>	Sejauh mana perangkat lunak memberikan perlindungan dari pengungkapan data atau informasi yang tidak sah, baik disengaja atau disengaja.
	<i>Integrity</i>	Sejauh mana ketepatan dan kelengkapan aset dijaga.
	<i>Non-repudiation</i>	Sejauh mana tindakan atau peristiwa dapat dibuktikan telah terjadi, sehingga peristiwa atau tindakan tidak dapat ditolak.
	<i>Accountability</i>	Sejauh mana tindakan suatu entitas dapat dilacak secara unik kepada entitas.
	<i>Authenticity</i>	Sejauh mana identitas suatu subjek atau sumber daya dapat dibuktikan sebagai yang diklaim.
	<i>Security compliance</i>	Sejauh mana produk perangkat lunak mematuhi standar, konvensi atau

		peraturan yang berkaitan dengan keamanan.
--	--	---

Sumber (Kurniawan, Arifianto and Muharom, 2018)

2.5. Skala Likert

Menurut Suryani (2015) *Skala Likert* digunakan untuk mengukur sikap, persepsi dan pendapat seseorang atau kelompok orang tentang sebuah fenomena sosial. *Skala Likert* dapat memberikan alternatif jawaban dari soal instrumen dengan gradasi dari sangat positif hingga sangat negatif, pertimbangan pemilihan pengukuran ini karena memudahkan responden untuk memilih jawaban Data kualitatif diubah berdasarkan bobot skor satu, dua, tiga, empat, dan lima. Jadi peneliti menggunakan *skala likert* untuk menilai sejauh mana responden setuju atau tidak setuju.

$$\text{Persentase Kepuasan (100\%)} = \frac{\text{Skor yang diobservasi (aktual)}}{\text{skro yang diharapkan (ideal)}} \times 100\%$$

Sumber : Suryani (2015)

Berikut ini tabel *Skala Likert* dan bobot skor disajikan dalam tabel:

Tabel 2. 10 *Skala Likert*

No	Kategori	Skor
1	Sangat Setuju	5
2	Setuju	4
3	Kurang Setuju	3
4	Tidak Setuju	2
5	Sangat Tidak Setuju	1

Sumber (Sugiyono, 2018)

Hasil persentase digunakan untuk memberikan jawaban atas kepuasan pengguna dari aspek-aspek yang diteliti. Menurut Arikunto (2009) pembagian kategori kepuasan

ada lima skala. Skala ini memperhatikan rentang dari bilangan persentase. Nilai maksimal yang diharapkan adalah 100% dan minimum 0%.

Tabel 2. 11 *Skor Persentase*

No	Persentase (%)	Kategori Kepuasan
1	<21%	Sangat Tidak Puas
2	21% - 40%	Tidak Puas
3	41% - 60%	Kurang Puas
4	61% - 80%	Puas
5	81% - 100%	Sangat Puas

Sumber (Sugiyono, 2018)