BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian *System*

System penjelasan Kristanto dalam jurnal (Oktafiani & Yunita, 2018). *system* ialah komponen-komponen yang terhubung dengan yaig laiin, mulai dari proses masukan, mengelola hasil dari masukan setelah di proses maka akan menghasilkan hasil yang dibutuhkan.

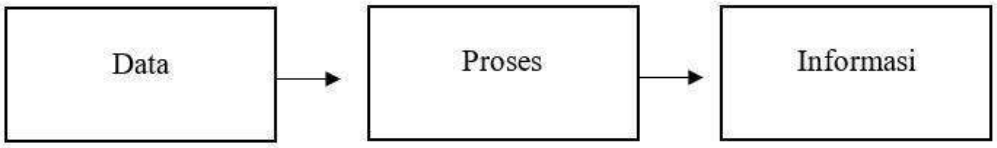
Sistem menurut (Agustin, 2018). Bisa artikan kumpulan *variable-variabel* yang saling berhubungan lalu berkomunikasi dengan yang lain serta saling membutuhkan satu sama lainnya untuk bisa tetap terhubung. *Sistem* setelah diracang di uji coba dan di *implentasikan* untuk kebutuhkan organisasi.

2.1.2 Pengertian Informasi

Penjelasan (Ridha, 2018). Informasi ialah dokumen atau file dapat digolongkan kemudian di olah untuk mengambil keputusan. Sistem akan mengelola dokumen atau file yang akan menghasilkan informasi atau berita yang sebelumnya dokumen atau file yang tidak bisa digunakan menjadi berguna dan bermanfaat setelah di olah menjadi satu kesatuan yang mengeluarkan hasil (*output*), kemudian informasi tersebut akan berguna bagi penerima.

Informasi menurut siregar dalam jurnal (Sihombing, 2018). Suatu *media* untuk memberikan informasi yang berguna untuk orang lain di kemudian hari.

Menurut *Mc Leoad* dalam jurnal (Agustin, 2018). Sistem adalah data yang sudah di olah sehingga bisa menghasilkan suatu informasi atau berita yang bisa digunakan dan bermanfaat sehingga bisa mengambil keputusan apa yang akan di lakukan.



Gambar 2.1 Data yang di olah yang menghasilkan

2.1.3 Pengertian Sistem Informasi

Menurut Anwar dijelaskan dalan jurnal (Sirin Mazaya Rochmah Shahab, Sirojul Munir, S.Si, 2019). Sistem informasi merupakan pengontrolan data, pengelolaan informasi yang saling terhubung untuk saling mengelola sehingga bisa menghasilkan suatu informasi atau berita yang bisa disampaikan untuk di jadikan berita atau informasi.

Menurut (Tukino, 2020) yang di kemukakan oleh Harumy et al.,2018. Sistem informasi ialah bentuk teknologi yang saling terhubung satu dengan yang lainnya dengan informasi, sehingga memudahkan dalam pengelolahan data dan informasi supaya bisa merubah data menjadi informasi yang bisa digunakan untuk organisasi atau orang yang membutuhkan.

2.1.4 *UML (Unified Modeling Languange)*

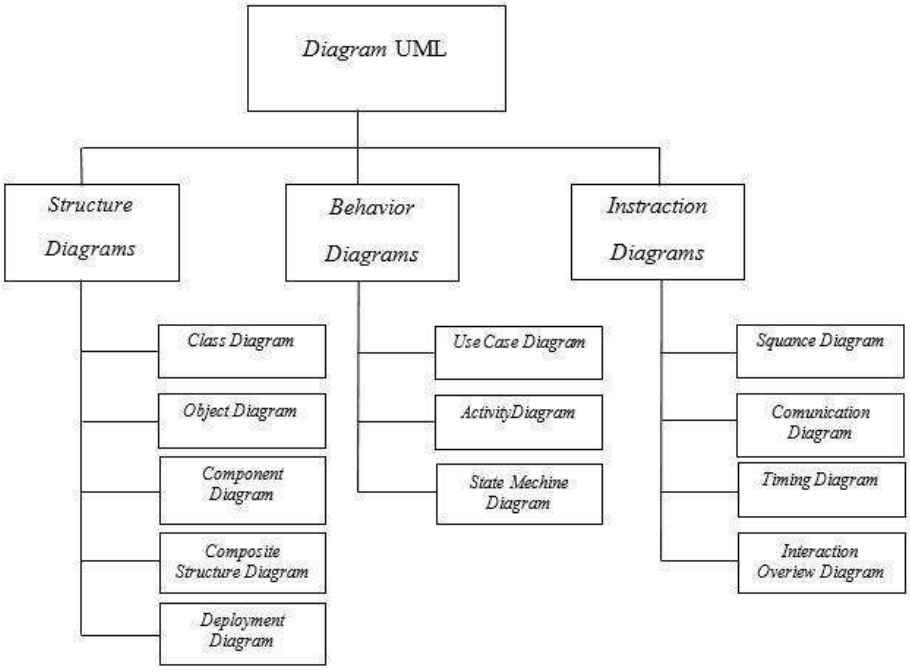
Menurut (Puspita & Anggita, 2020). UML ialah Teknik pemodelan visual yang digunakan dalam desain dan pembuatan sebuah *software* yang berorientasi *object*.

Pada buku tentang Merancang sistem berorientasi objek melalui pemodelan UML (Anardani, 2019). UML merupakan penyatuan dari pemodelan *Booch Method, Object Modeling Tecnique* (OMT) dan *Object Oriented Software Enginnering*.

a. Pemodelan *Boch* yang dikenal *dengan Design Object Oriented* yaitu identifikasi kelas objek, indentifikasi hubungan kelas dan objek, identifikasi antarmuka dan yang terakhir *Implementasi.*

b. Pemodelan OMT yang dipelopori oleh *Rumbaugh* dikembangkan dengan dasar analisis perancangan tersktruktur dan pemodelan *entity relationship.*

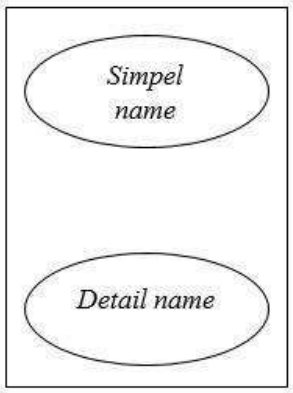
c. Metode OOSE milik *Jacobson* memiliki analisis kebutuhan perancangan dan yang terakhir yaitu implementasi dan pengujian. Lalu *diagram* UML mempunyai 13 jenis bagian yang mempunyai fungsi berbeda-beda yaitu :



Gambar 2.2 *Diagram uml*

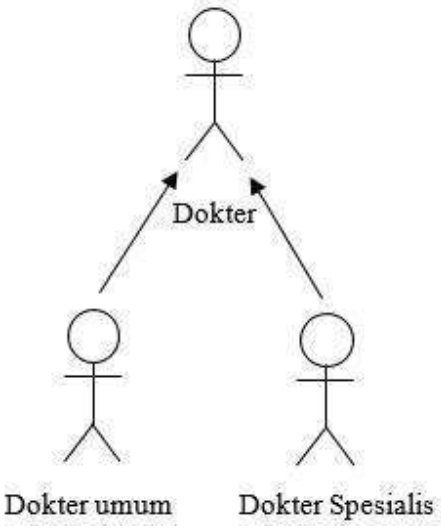
1. *Use Case Diagram*

Digunakan memvisualisasikan *system* terhadap *actor*, kemudian pada saat membuat *use case diagram* di timbal balikan kegunaanya dengan *actor* berupa sebuah fungsi sistem (Hatta et al., 2019).



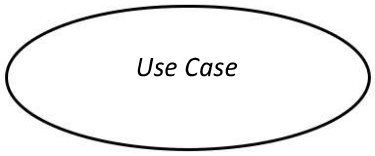
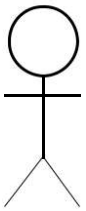
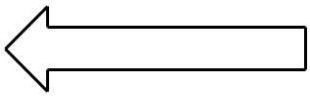
Gambar 2.3 Nama *Use Case*

Actor bekerja dengan mencari peran pendukung secara spesifik dari pengguna satu dengan pengguna lainnya dan saling berinteraksi dengan *use case*.



Gambar 2.4 Contoh a*ctor*

Tabel 2.1 Jenis-jenis relasi

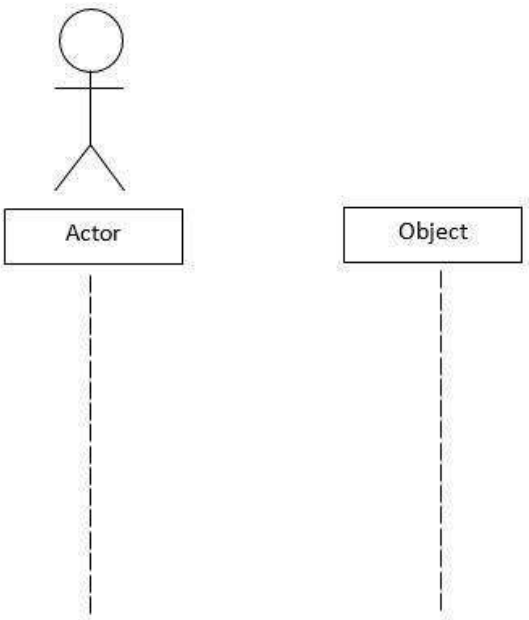
Simbol	Keterangan
Use Case 	Menjelaskan alur tindakan dilakukan oleh *system* untuk menghasilkan aktor yang terukur.
Aktor / *Actor* 	Tentukan peran yang dimainkan pengguna saat berinteraksi dengan *use case*.
Asosiasi	Hubungan berbagai objek satu sama lain sehingga terhubung satu sama lain.
Includ/ *Include*	Menentukan kasus penggunaan sumber bersifat eksplisit.
Generalisasi / *Generalization*	Hubungan objek(keturunan) berbagi prilaku dan stkruktur data objek pada objek utama(slice).
Extensi / *Extend* 	Tentukan prilaku kasus penggunaan tartget untuk memperluas Kasus penggunaan untuk sumber daya pada titik waktu tertentu.
Sistem 	Menampilkan paket dari system tertentu.

2. *Sequence Diagram*

Menurut (Syarif & Nugraha, 2020), adalah gambaran UML yang saling terhubung antar objek yang berada di sekitar sistem, sedangkan menurut (Hutabri & Putri, 2019). *Sequence diagram* menjelaskan tingkah laku objek *use case* dengan memaparkan tujuan waktu dari objek *message* yang di *receive* kemudian *send* antar *object*. Berikut merupakan urutan pengiriman pesan antar *object*.

Notasi *Sequence diagram*

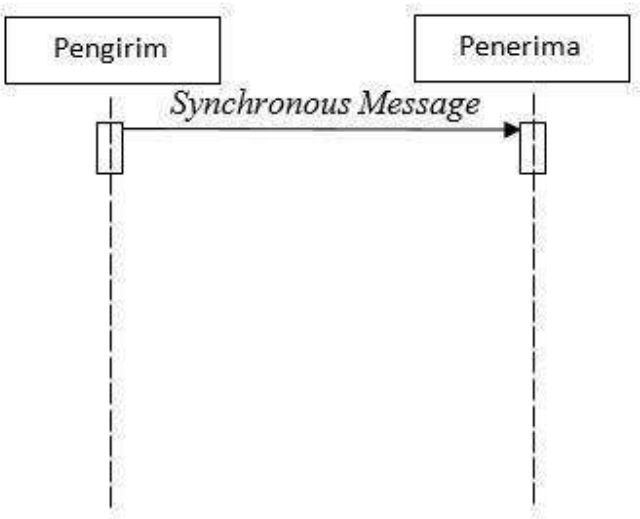
a. *Lifeline stereotype icon*, komponen yang mengeksekusi mulai dari pesan dikirim maupun pesan di terima.



Gambar 2.5 *lifeline stereotype*

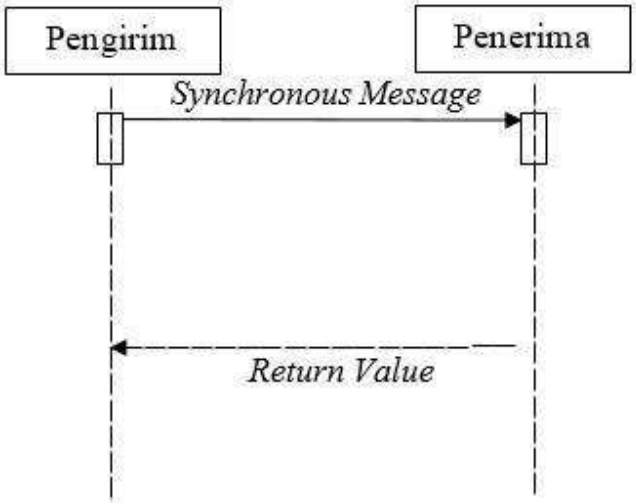
b. *Messages* merupakan kumpulan yang berhubungan dengan objek yang menjelaskan langkah apa yang akan dilakukan.

c. *Sync messages* bisa di ambil contoh pengirim ingin mengirimkan pesan kepada penerima proses ini membutuhkan proses yang dimana pengirim sudah membuat pesan dan siap dikirim ke penerima maka penerima baru bisa menerima pesan yang dikirim oleh penerima.



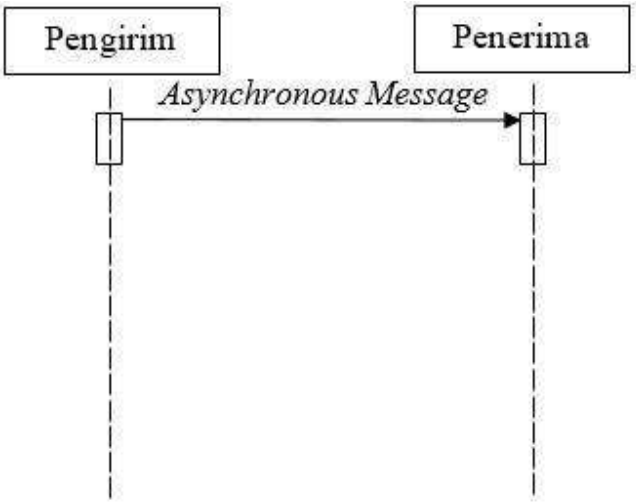
Gambar2.6*Synchronous Message*

d. *Synchronous Message dengan return value* adalah penerima pesan dapat pengirimkan balasan pesan kepada pengirim dengan gambaran garis putus-putus yang ditampilkan pada gambar berikut.



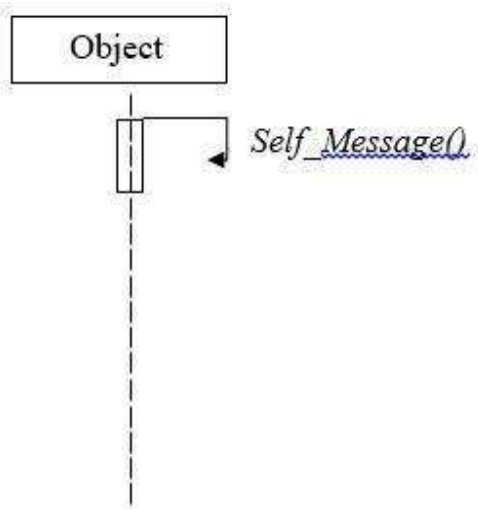
Gambar 2.7 *Syncronous Message return value*

Asynchronous Message pengirim hanya bertanggung jawab pengirim pesan ke penerima dan tidak menunggu hingga proses selesai di terima oleh penerima.



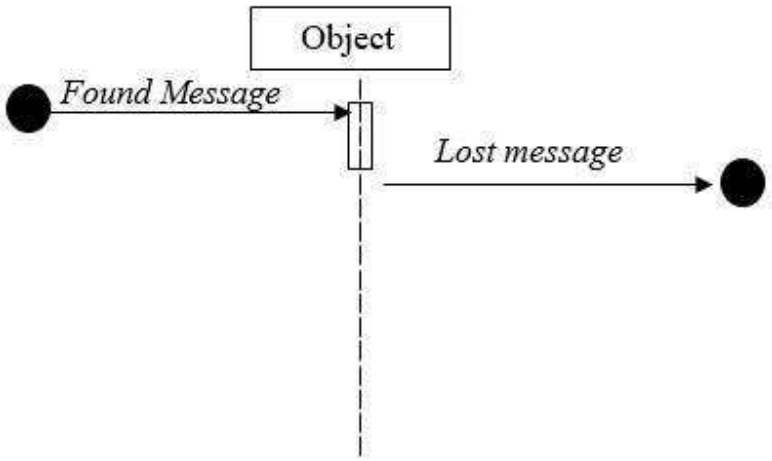
Gambar 2.8 *Asynchronous message*

e. *Self Message* merupakan objek memanggil dirinya.



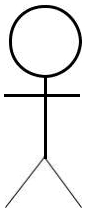
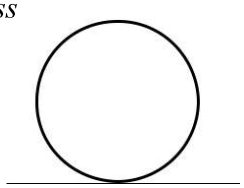
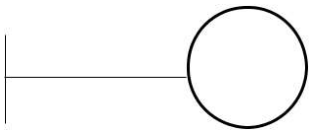
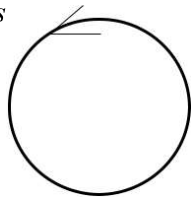
Gambar2.9*Self Message*

f. *Lost and Found Message,* pesan yang hilang adalah pesan yang dikirim tetapi pesan tidak sampai ke penerima yang di tuju.



Gambar 2.10 *Lost and found message*

Tabel 2.2 *Sequence diagram*

Simbol	Keterangan
Aktor 	Menggambar pengguna yang berhubungan terhadap *system*
Entity Class 	Memvisualisasikan yang akan dilakuakn
Boundary Class 	Menjelaskan desain formulir.
Control Class 	Menjelaskan hubungan antara batas dan tabel.
A focus of control & a life line 	Jelaskan lokasi serta tujuan pesan.
A Massage 	menjelaskan proses *sending* message.

3. *Activity Diagram*

Yang jelaskan (Syarif & Nugraha, 2020). *Activity diagram* merupakan gambaran aliran aktifitas *system* merupakan gambaran alur bisnis dapat di gambar pada *software enginner* (*Software*) pendukung.

Tabel 2.3 aktivitas *diagram*

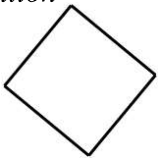
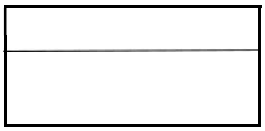
Simbol	Keterangan
Aktivitas 	menjelaskan bagaimana antarmuka kelas berkomunikasi.
Aksi 	Keadaan *system* dapat memvisualkan pelaksanaan suatu tindakan.
Initial Node 	Proses memulai.
Activity Final Node 	Dibentuk ataupun dihancurkan.

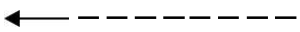
Fork Node 	di titik menjadi aliran banyak aliran.

4. *Class Diagram*

Class diagram menurut (Hutabri & Putri, 2019). Kelas yang dipakai membangun sistem serta menjelaskan langkah sistem definisi kelas, varibel dan fungsi-fungsi digunakan oleh *class diagram.*

Tabel 2.4 kelas *diagram*

Simbol	**penjelasan**
Generalization	Dimana *object* (keturunan) berbagi *structure file* objek di atas objek induk (aksesor).
Nary Association 	Menjelaskan hubungan 2 keputusan.
Class 	Sekumpulan *object* memiliki petunjuk dan operasi yang sama.

Realization 	Penjelasan *return* dilakukan *object*.
Dependency 	Perubahan relasi pada *elemen indepen* dapat mempengaruhii *elemen* yang terhubung pada *dependen elemen.*
Association	Menjelaskan antar *object* satu dengan *object* lainnya
Collarboration 	Deskripsi rangkaia tindakan yang diproses system untuk hasil.

2.2 Teori Khusus

2.2.1 Penjelasan Kas

Pengelolaan dana kas yang dilakukan oleh ketua RT maupun RW harus diterapkan berdasarkan perioritas pengguna, uang kas digunakan untuk membangun dan mensejahterakan warga (Hidayah & Wijayanti, 2017).

Menurut (Rasminto et al., 2019). Pengelolaan adalah membuat susunan kegiatan seperti desain, pelaksanaan dan pengawasan supaya bisa menuju ke tujuan yang sudah di gambarkan sebelumnya.

2.2.2 Perincian Anggaran

Menurut Munir yang dijelaskan dalam jurnal (Khaeirudin, 2016). Anggaran adalah tujuan kinerja yang akan di capai beberapa bulan kedepan atau periode yang di targetkan dalam segi keuangan, anggara yang dimiliki oleh RT atau RW pemprosesan dana harus memperhatikan gambaran keterbukaan satu dengan yang lainnya. Warga atau masyarakat harus mengetaui rincian pemakaian dana anggaran apa saja yang sudah digunakan maupun yang belum digunakan.

Rekap data keuangan menurut (Dewi Kirowati & Vaisal Amir, 2019). Merupakan bagian dari siklus akuntasi mulai dari *transaction*, pencatatan bukti, Jurnal, Buku besar, Neraca penyesuaian, rekap data keunagan, pembukuan penutup, Neraraca saldo setelah jurnal penutup.

2.2.3 Penelitian terdahulu

Tabel 2.5 *Survey* sebelumnya

No	Nama pengarang dan tahun	Keterangan
1	*System* pelaporan keuangan RW berbasis *web* (Tlali et al., 2019).	Aplikasi yang dibangun sudah bisa mempermudah pencatatan kas keluar dan masuk.
2	Merancang dan membangun dashbor sistem cerdas untuk sistem manjemen RT mendukung masyarakat 5.0 (Sony & Sabaruddin, 2020).	Hasil penelitian menunjukan aplikasi *smart system* RW dan RT bisa menggantikan pencatatan dan pengelolaan data manual seperti pencatatan kas keluar dan masuk, administrasi.

3	Analisis dan Perancangan Aplikasi Layanan RT dan RW Cerdas Publik Kecamatan Sindang Kecamatan Sindang Desa Kanal (Bani Muhamad et al., 2020).	Penelitian secara umum berhasil dilakukan namun untuk pelaporan duplikasi masih mengalami kendala sehingga pelaporan mengalami duplikasi.
4	Sistem Akuntansi Kas buat Transparansi Keuangan (Studi Kasus Pembangunan Masyarakat Desa Kradon RT.003 dan RW.043 Malangan)(Purnomo & Rozi, 2020).	Hasil penelitian yang sudah dilakukan memberikan kemudahan dalam proses pembukuan kas RT dan RW yang meliputi pencatatan dana keluar dan masuk.
5	Memberikan konsultasi Rencana keuangan keluarga islami warga RT KuncenSukoharjo (Al-Hakim et al., 2020).	Kegiatan penelitian penyuluhan perencanaan keuangan keluarga islam sulitnya dalam menerima materi yang diberikan, padahal menurut civitas materi tersebut tergolong mudah oleh sebab itu harus harus dibuat kembali materi yang lebih simple agar mudah di terima oleh warga.
6	Perancangan sistem informasi RT dan RW perumahan Swan *Regency*	Hasil dari penelitian yang sudah dilakukan yaitu membantu warga RT dan RW pada

	berbasi perangkat bergerak (Syahrizal Setiawan Wicaksana, 2021).	perumahan swan *regency* mengelola keuangan.
7	Gunakan teknologi skala kegunaan sistem untuk mengevaluasi tingkat *utilitas* aplikasi managemen populasi (Ependi et al., 2019).	Kegiatan penelitian memiliki kegunaan yang sangat memuaskan sesuai dengan minat warga.
8	Penerapan simade aplikasi (*information system* pengelolaan) pelayanan administrasi Bagan Sinembah, Jalan raya Kepenghulu bakti dan Riau hilir Kab.Rokah (Sihombing, 2018).	Dari penelitian yang dilakukan maka hasilnya adalah mempermudahkan dalam proses laporan dan pembuatan surat menyurat serta bisa melihat grafik data penduduk.
9	Desain beroritasi objek dan peembangunan sistem terpadu untuk pengelolaan dan penggunaan dana di tingkat Desa.(Rasminto et al., 2019).	Hasil dari penelitian yang sudah dilakukan mempermudah dalam penyusunan anggaran pendapat dan belanja desa.
10	Menerapkan Instansi pemerintah kota Desa Suwon, Distrik Karawatto, pene Minahasa, utara Prefektur	Dari penelitian yang sudah dilakukan (mis. rapan untuk Sistem baru bekerja Prefektur

	Minahasa) *System* keuangan desa (siskeudes)(Malahika et al., 2018).	dengan baik dalam prosedur perencanaan dan persiapan. pelaksanaan dan laporan akhir.
11	Menganisis peran Sistem keuangan desa (siskeudes) diterapkan untuk meningkatkan kualitas pertanggungjawaban keuangan kota dari perspektif ekonomi syariah (Ridwan, 2019).	Penelitian yang sudah teleti maka dengan di buatnya sistem aplikasi keunagan desa maka pencatatan *finance* menjadi lebih mudah, pelaporan anggaran keuangan menjadi lebih cepat dan akurat.
12	Analisis *usability* mengukur efektivitas penerapan sistem keuangan tingkat Desa (Sulindawati, 2018).	Hasil penelitian yang dilakukan respon dari penggunaan sistem keuangan desa yaitu cukup puas karena mempermudah dalam pengelolanan keuangan desa.
13	*Rural transfers: lessons learned from community development plans and rural funds (Watts et al., 2019).*	*Funds from Althougt village can be used to finance conservation and reforestation activities, and it is unlikely that the community will choose funds for this purpose. When multiple options are available*
14	*Smart innovation in 5G network-based rural finance and the Internet of Things (Cheng, 2020).*	*Village funding can be allocated specifically for conservation and reforestation at the community level, based*

		on lessons learned from the PNPM-LMP Program.
15	*Coordination of supply customer domains: take smart village and paper control in rural network contender for example Villages in the Regional Smart (Prinsloo et al., 2017).*	*The transactive features described in this approach a strong technology transaction 485 orchestrating the micro-network as an Energy Environment Network for equipment supply and operating load regulation.*

2.2.4 *Website*

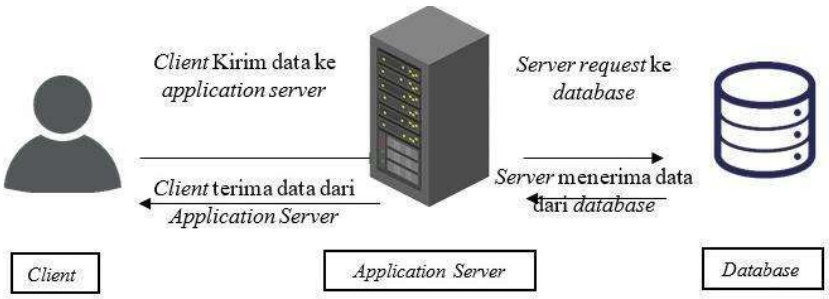
Dalam sebuah studi yang dilakukan (Erlin Elisaa et al., 2020) bertemakan perancangan sistem informasi jasa bantu pindah berbasis *Website* dapat di uraikan yang terdiri dari elemen-elemen halaman yang berguna memperbaiki sistem supaya bisa berjalan dengan lancar.

Menurut buku tentang penjelasan, evaluasi strategi membangun *Web* Perspektik rancang yang di jelaskan (Abdurrahman Sidik, S.Sn., 2019 : 14). *Website* atau situs merupakan sekumpulan halaman yang berisi informasi berbentuk digital. Informasi tersebut bisa seperti tulisan, *picture*, suara, vidio, animasi. Bisa buka di semua Dunia ketika memiliki akses perambahan.

2.2.5 *Database*

Database menurut (Mulyodipuro, 2018). Merupakan pendukung utama disebuah *system* informasi sebab sangat berguna untuk tempat *storage file* yang akan di kelola, *database* bisa penting karena bisa mengelola *document* menghindari ke sama data (duplikat), lalu untuk proses pengambilan data dari penyimpanan memerlukan perangkat

lunak pendukung lainnya seperti *database management system* (DBMS) untuk bisa mengelola data, mengambil data serta merubah dan menghapus data.



Gambar 2.11 *Database management system*

2.2.6 *MySQL*

Menurut (Eko Budi Setiawan, S.Kom & Angga Try Ramdany, 2019: 5). *MySQL* bisa dikatakan tempat atau wadah dalam meyimpanan database, yang mendukung Bahasa *database SQL (Structured Query Language). MySQL* ialah *software DBMS (Database Management System)* yang bisa *multithread multi user*.

MySQL ialah salah satu konsep kunci dari *database.* sejak dahulu, merupakan operasi *database* konsep, dalam mengelola data, seleksi terutama dalam proses pemasukan data, kecepatan dan ke handalan suatu system *database* (DBMS) bisa di ketahui mulai *system* kerja serta *optimasi* seperti penggunaan syntak perintah mysql. *MySQL* dapat nilai sangat *user friendly* dengan *database* seperti hal pencarian berkas, berikut adalah kelebih dari *MYSQL:*

1. *MySQL* bisa *running* di macam-macam *operating system*.
2. *MySQL* merupakan *database opensource.*

3. *MySQL* bisa dipakai beberapa pengguna pada waktu bersamaan tanpa terjadinya tabrakan data.
4. *MySQL* mempunyai *speed* akses bisa mempermudah proses pencarian data.
5. *MySQL* memiliki *interface* (antarmuka) *API (Application Programming Interface).*

2.2.7 PHP (*Hypertext Preprocessor*)

Ialah salah satu *open source* yang bisa terhubung dengan HTML, pada awalnya PHP atau bisa di sebut (Situs Personal) (Eko Budi Setiawan, S.Kom & Angga Try Ramdany, 2019).

Php dibuat Rasmus Leardorf pada tahun 1995 pertama kali, semasa itu mempunyai nama *Form Interpreted* (FI), mengelola web data formulir. Setelah itu menerbitkan *code* umum serta menambahkan FI/PHP, berikut ini adalah kelebihan Bahasa pemgrograman PHP:

1. *Server web* banyak *support* terhadap php dan mudah untuk proses pengaturan.
2. PHP tergolong dapat dipahami, karena banyak programmer dapat mengembangkan.
3. Sangat mudah untuk di pahami sumber bacaan sangat banyak
4. Bahasa pemrogramana PHP dapat di masukan ke dalam HTML.
5. Cocok digunakan dalam pemrograman web dinamis.
6. Bahasa pemrograman terbuka PHP, dapat di pakai di berbagai *operating system* apapun.

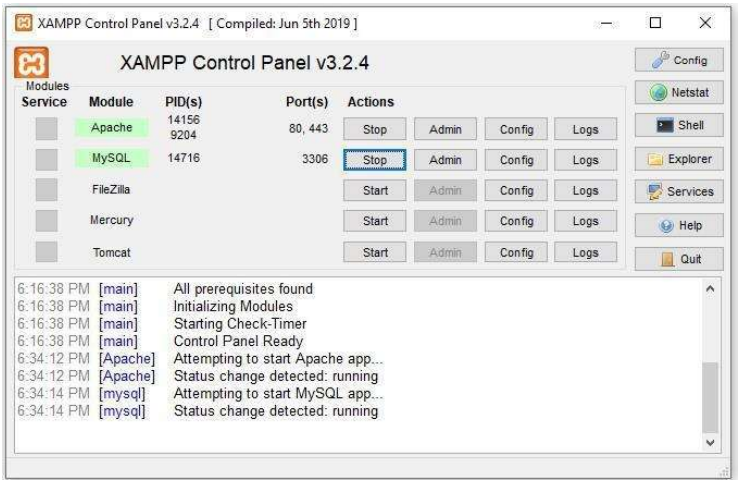
2.2.8 *XAMPP*

Menurut *Madcoms* dalam jurnal (Fitri Ayu and Nia Permatasari, 2018), *xampp* sekumpulan paket antara lain yaitu *Apache, MySQL*, *PhpMyAdmin, PHP, Perl, Filezilla*,

untuk membangun suatu aplikasi kita membutuhkan beberapa *software* pendukung untuk mempermudah dalam membuat program atau aplikasi, *software* pendukungnya seperti:

1. *Xampp* sebagai *service server* untuk php dan *MySQL*.
2. *Codeingniter* sebagai *Framework* untuk memudahkan dalam membuat program.

Untuk memastikan aplikasi *xampp* berjalan dengan lancar kita cari *xampp control* setelah itu kita buka aplikasi *xampp control* dengan menekan *double* klik hingga *menu xamp control* muncul, setelah itu kita klik *start Apache* dan *MySQL.* Apabila tidak ada kendala seharusnya *Apache* dan *Myl* akan *running*.



Gambar2.12 *Control panel activation*

2.2.9 *Codeingniter*

Menurut (Rofiah, 2018). *Codeigniter* merupakan php *framework* dapat dipakai serta mempermudah *scriipt* serta mengoptimasi. Cara model kerja MVC berguna memudahkan dalam membuat *web* sederahan dengan memakai php.

Model View Control yang di jelaskan (Sirin Mazaya Rochmah Shahab, Sirojul Munir, S.Si, 2019). Merupakan gambaran yang cukup terkenal dalam pembuatan *website*. MVC

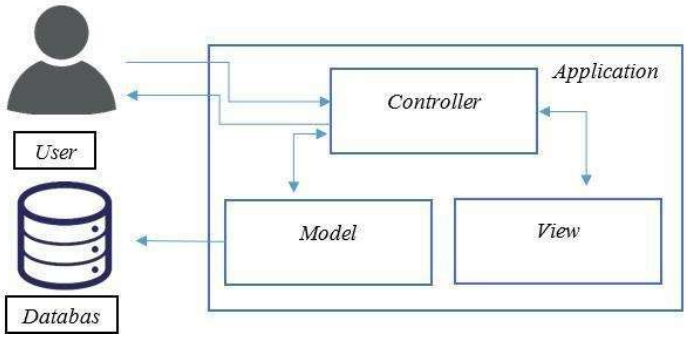
menggolongkan memanipulasi struktur aplikasi data, tampilan antarmuka, dan bagian pengontrolan aplikasi ini menjadi 3 penjelasan yaitu seperti berikut :

1. Komponen MVC

a. *Model* terhubung langsung ke *database (insert,update,delete)* dan biasanya menagani validasi di bagian *controller*, tapi tidak langsung di *view.*

b. tampil bertugas menangani *presentation logica,* yang berupa kode-kode untuk di tampilkan.

c. *Controller* ialah bagian mengontrol relasi model dan bagian tampil, *controller* bertugas penerima permintaan berkas pengguna lalu memilih yang akan di kelola aplikasi.

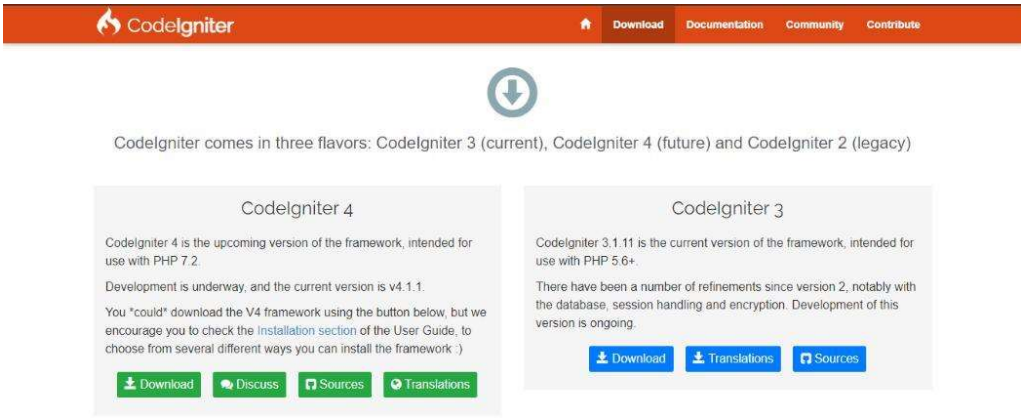


Sumber : Subagia,2018

Gambar 2.13 Cara Kerja *MVC (model,view,control)*

2. Installasi Codeigniter

Untuk proses *installasi codeigniter* sangat mudah, sebelum masuk ke proses instalasi codeigniter *download* terlebih dahulu dengan mengujungi situs https://codeigniter.com/download, pada *link* tersebut ada 2 pilihan *codeigniter* ingin menggunakan versi 3 atau versi 4, berikut adalah tampilan gambar seperti dibawah ini.

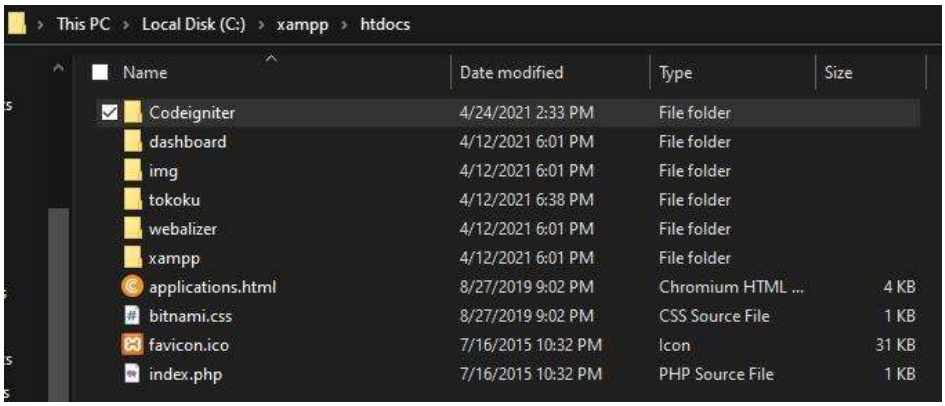


Gambar 2.14 Pilih versi *Codeigniter*

Untuk proses *installasi codeigniter* versi 3 langkah pertama terlebih dahulu kita men*download* filenya, jika suda selesai men*download* langkah berikutnya yaitu dengan cara meng *extract* file jangan lupa untuk mengganti nama menjadi *codeigniter.*

Gambar 2.15 *file* yang di butuhkan

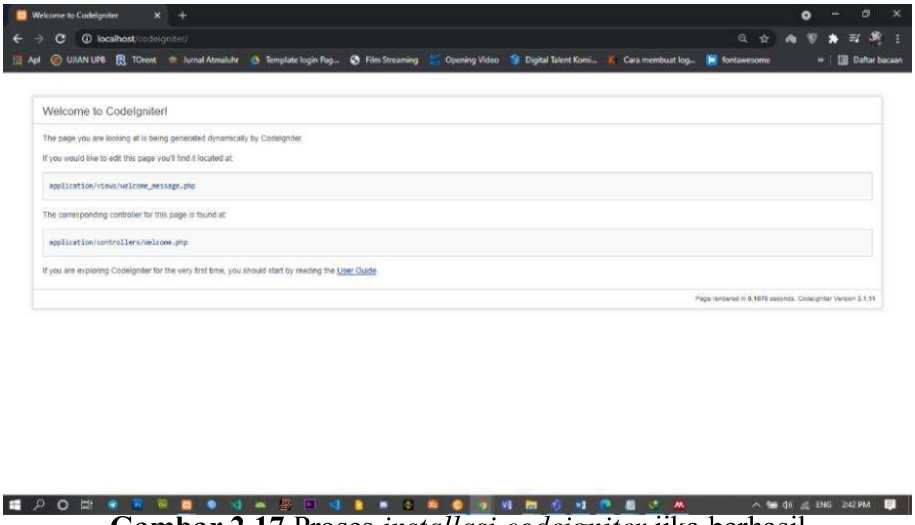
Jika sudah selanjutnya masuk ke tahap berikutnya dengen mencopy hasil *extract* tersebut pada *directory* C:\xampp\htdocs seperti pada gambar berikut.



Gambar 2.16 *directory* penyimpnan *codeigniter*

Langkah berikutnya yaitu mengkonfirugrasi pada *directory* : C:\xampp\htdocs\Codeigniter\application\config\. Apabila sudah berada directory ini kita buka file *config.php* bisa menggunakan notepad ataupun teks editor seperti *visual studio code*. Lalu cari file berikut *$config ['base_url'] = '';* apabila sudah ditemukan maka ubah menjadi *$config ['base_url'] = 'http://localhost/codeigniter';*. Apabila sudah kita buka *mozile firefox* ataupun

google chrome kita ketik http://localhost/codeigniter/ proses installasi sudah selesai.



Gambar 2.17 Proses *installasi codeigniter* jika berhasil

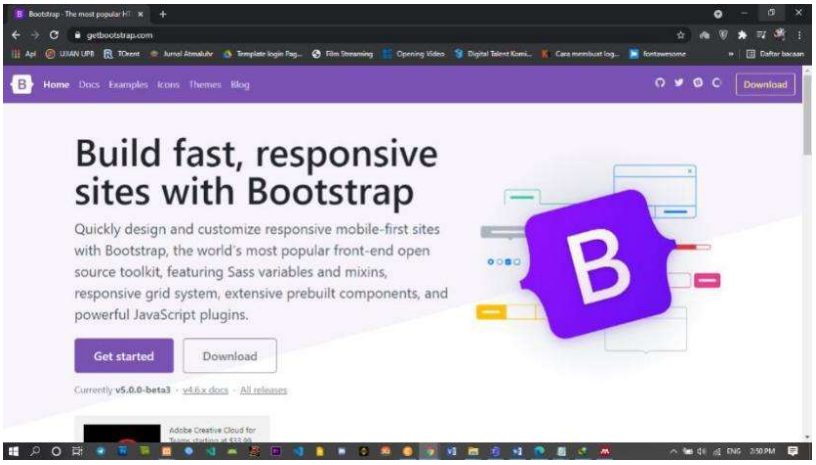
2.2.10 *Bootsrap*

Bootstrap adalah pendukung untuk membuat tampilan halaman *website* dapat mempermudah mebangun sebuat *website* bagi pengembang maupun bagian yang membuat desain *website* tersebut. Tujuan Subardio dalam jurnal (Basuki, 2019). Seorang pengembang *website* dan spesialis ICT dapat menerapkan konsep atau pola piker menggunakan MVC (*Model, View, Control*), tujuan dari menggunakan *bootstrap* ialah untuk menghindari kesalahan dari setiap ukuran *device (flatform).* Selain itu *bootsrtrap* mempermudah pengembang dalam membuat *website* yang dimana semua komponen sudah di pisahkan dan

dapat digunakan sesuai dengan kebutuhan, pada saat melakukan perubahan tidak kesulitan karna komponen tersebut sudah berada pada sub – sub tertentu sebagai contoh *Model,View* dan *Control* sudah dipisahkan karena untuk mempermudah dalam memasukan *source code* dan pencarian file ketika terjadi *error.*

1. *Installasi Bootstrap*

Untuk proses *installasi bootstrap* pertama kita harus men*download* terlebih dahulu di *website* : https://getbootstrap.com/



Gambar 2.18 *Boostrap*

Setelah selesai mendownload *file bootstrap* langkah selanjutnya yaitu meng *extrak* dan *copy* ke dalam *website* yang akan di bangun atau di gunakan.

2. Struktur *Bootsrtrap*

Menurut (Santoso, 2019), *bootstrap* menggunakan *grid system*. Untuk megubah *size* tampilan antar muka yang berfungsi mengatur tampilan yang lebar menjadi sesuai dengan ukuran *devices* yang digunakan, berikut ini beberapa *grid system* dari bootsrap seperti dibawah ini.

a. *Meta Tag name viewport* merupakan perintah kepada *browser* untuk menampilkan tampilan sesuai dengan perangkat bisa di katakana sebagai *responsive* yang dimana bertujuan untuk berselancar atau mengoptimalkan berbagai perangkat sehingga tampilan antar muka *website* dapat sesuai dengan *devices* apa yang digunakan.
b. *Bootsrtrap css* adalah file utama css untuk membuat modifikasi seperti ukuran tulisan, *font*, memberikan warna *table* dan masih banyak yang bisa dilakukan css.
c. *Jquery.Js* merupakan file yang berfungsi untuk mejalankan fitur-fitur seperti membuat *navigasi, slider, dropdown* agar tampilan terlihat lebih enak untuk dilihat.