

BAB II

TINJAUAN PUSTAKA

2.1 Repository

Repository atau gudang sangat di butuhkan untuk kepentingan penyimpanan atau dokumentasi suatu data, sehingga pada saat data di butuhkan suatu-waktu dapat di pergunakan kembali sebagaimana mestinya. Repository adalah struktur data yang menyimpan metadata untuk sekumpulan struktur berkas atau direktori. Bergantung pada apakah sistem kendali versi yang digunakan itu didistribusikan (misalnya, Git atau Mercurial) atau terpusat (misalnya, Subversion atau Perforce), seluruh rangkaian informasi dalam repositori dapat diduplikasi pada sistem setiap pengguna atau dapat dipelihara pada satu peladen. Beberapa metadata yang ada di repository antara lain adalah:

1. Catatan riwayat perubahan dalam repositori.
2. Sekumpulan objek komit.
3. Kumpulan referensi untuk objek komit yang disebut heads.

Dengan kata lain repository ini adalah paket-paket khusus untuk sebuah sistem operasi yang kemudian paket-paket tersebut di instal untuk mendapatkan kinerja lebih baik dari sebuah sistem operasi. Perguruan tinggi yang berbasis repository adalah satu set layanan yang menawarkan berbagai bahan digital yang dihasilkan oleh lembaga tersebut ataupun yang dihasilkan lembaga lain yang dikelolanya kepada masyarakat penggunaanya (Pfister 2008).

2.2 Application Programming Interfaces (API's)

Application Programming Interface (API) bukan hanya *signature* yang sederhana, dan satu *set class* atau *method* fungsi API yang memiliki tujuan utama untuk mengatasi ketidak pahaman dalam membangun *software* yang berukuran besar, mulai

dari yang sederhana sampai kompleks serta komponen yang sulit dipahami (Halim, Kevin, and Stevaea 2011).

2.3 Quasar Framework

Quasar adalah framework JavaScript yang bisa digunakan untuk mengembangkan aplikasi single-page yang bisa menggabungkan paket program JavaScript dengan komponen tunggal yang dikembangkan. Framework ini menawarkan lingkungan pemrograman yang komprehensif untuk pengembangan aplikasi modern nan responsif. Quasar tersusun dari blok bangunan tingkat tinggi yang disebut komponen. Komponen ini bisa digunakan untuk membuat antarmuka aplikasimu dengan cepat.

2.4 Database

Sistem basis data merupakan sistem yang terdiri dari atas kumpulan tabel yang saling berhubungan (dalam sebuah basis data disebut sistem komputer) dan sekumpulan program (yang biasa disebut DBMS atau *DataBase Management System*) yang memungkinkan beberapa pemakaidan/atau program lain untuk mengakses dan memanipulasi tabel-tabel data tersebut.

2.5 Visual Studio Code

Visual Studio Code merupakan sebuah kode editor yang dikembangkan oleh *microsoft* untuk *windows*, *linux* dan *macOs*. Proram ini termasuk dukungan untuk debugging, kontrol git yang tertanam dan *GitHub*, penyorotan sintaktis, penyelesaian kode cerdas, *snippet* dan *refactoring* kode, ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, penyintasan *keyboard*, *preferensi* dan meninstall ekstensi yang menambah fungsional tambahan.

2.6 NoSQL

NoSQL *database* bisa bekerja secara cepat dibanding basis data relasional. Perkembangan *web* yang cukup cepat menimbulkan kemajuan NoSQL karna selaku alternatif dalam memesatkan akses dibandingkan memakai basis informasi relasional.

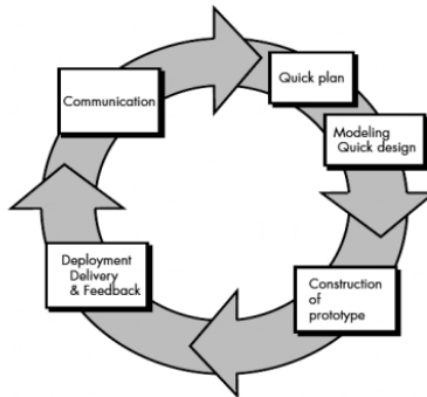
Ukuran kecepatan di dalam NoSQL ialah salah satu topik yang sangat menarik dalam dikaji untuk suatu riset. Sebagian riset yang akan terhubung dengan ukuran kecepatan, perbandingan performa sempat dicoba dalam riset sebelumnya, antara lain:

1. Perbandingan performa nosql dengan relational *database*.
2. Perbandingan kinerja relational *database* dengan data *manipulation language* nosql.

Riset perbandingan konsep relational *database* dengan nosql. Bagi survey, peringkat awal dalam urutan *database* NOSQL berbasis *stored document* adalah MongoDB (Gunawan 2018).

2.7 Prototype

Prototyping perangkat lunak (*software prototype*) atau siklus hidup menggunakan *prototyping (life cycle using prototyping)* yaitu salah satu metode siklus hidup sistem yang didasarkan pada konsep model bekerja (*working model*). Tujuannya untuk mengembangkan model menjadi sistem *final*. Artinya sistem akan dikembangkan lebih cepat dari pada metode tradisional dan biaya menjadi lebih rendah. Ada banyak cara untuk membuat sebuah *prototype*, begitu juga dengan penggunaannya. Ciri khas dari metode ini adalah pengembang sistem, klien dan pengguna dapat melihat dan melakukan eksperimen dengan bagian dari sistem komputer sejak awal proses pengembangan. Metode Prototype merupakan metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna (Pressman, 2012: 50). yang dapat dilihat pada Gambar 2.2.



Gambar 2. 1 Metode Prototype Roger S. Pressman, 2012:50

1. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak dengan mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat.
2. Membangun *prototype*

Membangun *prototype* dengan membuat perancangan sementara untuk penyajian kepada pelanggan.
3. Evaluasi *prototype*

Evaluasi ini dilakukan oleh pelanggan apakah *prototype* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototype* direvisi dengan mengulang langkah 1, 2 dan 3.
4. Mengkodekan sistem

Dalam tahap ini, *prototype* yang sudah disepakati akan dimasukkan ke dalam bahasa pemrograman yang sesuai.
5. Menguji sistem

Setelah sistem menjadi perangkat lunak yang siap pakai, terlebih dahulu sistem diuji sebelum digunakan. Pengujian ini dilakukan dengan *white box* dan *black box* berbasis path pengujian arsitektur dan lain-lain.
6. Evaluasi sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sesuai dengan yang diharapkan. Jika iya, maka langkah 7 dilakukan, jika tidak, maka langkah 4 dan 5 dilakukan.

7. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

2.8 Website

Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (hypertext transfer protokol) dan untuk mengakses menggunakan perangkat lunak yang disebut browser (Arief : 2017). Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman (Hidayat : 2010:2).

2.9 Unnified Modeling Language (UML)

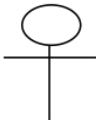
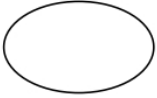


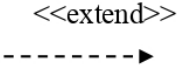
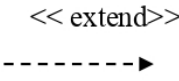
UML berkembang pesat pada akhir 1980 -an dan awal 1990-an UML lahir dari penggabungan banyak bahasa permodelan grafis yang berorientasikan objek. UML dibuat oleh James Rumbaugh, Ivar Jacobson dan Grady Booch. di bawah bendera *Rational Software Corp.* UML memberika notasi-notasi yang dapat membantu memodelkan sistem dari berbagai perspektif. UML juga tidak hanya dapat digunakan dalam pemodelan perangkat lunak, hampir dalam semua bidang yang membutuhkan pemodelan (Fowler, 2003).

1. Use Case Diagram

Menyajikan interaksi antara *usecase* dan *actor* merupakan diagram *usecase*.

Contoh diagram *usecase* disajikan pada Gambar 2.3 (John W. Satzinger, 2016).





Tabel 2. 1 Usecase

Notasi	Keterangan	Simbol
<i>Actor</i>	Pengguna sistem atau yang berinteraksi langsung dengan sistem, bisa manusia, aplikasi, ataupun objek lain.	
<i>Use case</i>	Fungsionalitas yang disediakan sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.	
<i>Association</i>	Digambarkan dengan sebuah garis yang berfungsi menghubungkan actor dengan <i>Use case</i> .	
<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>Use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.	
<i>Includes</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, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya <i>Use case</i> tambahan memiliki nama depan yang sama dengan yang ditambahkan.	

2. Activity Diagram

Activity diagram ataupun Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Pada sesi pemodelan bisnis, diagram kegiatan bisa digunakan untuk menggambarkan aliran peristiwa (*flow of events*) dan membuktikan aliran kerja bisnis (*business work flow*) dalam *usecase*. Contoh diagram aktivitas disajikan pada Gambar 2.4 (John W. Satzinger, 2016).

Tabel 2. 2 Activity Diagram

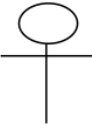






Notasi	Keterangan	Simbol
<i>Initial State</i>	Titik awal memulai aktivitas.	
<i>Final State</i>	Titik akhir mengakhiri aktivitas.	
<i>Activity</i>	Menandakan sebuah aktivitas.	
<i>Decision</i>	Pilihan pengambilan keputusan.	
<i>Fork / Join</i>	Menunjukkan kegiatan menggabungkan dua panel <i>activity</i> menjadi satu atau satu panel <i>activity</i> menjadi dua.	

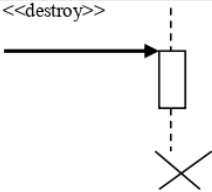
3. Sequence Diagram

Sequence diagram atau Diagram sekuensial biasa digunakan untuk menunjukkan aliran fungsionalitas dalam *usecase*. Contoh pada *usecase* “menarik uang” mempunyai beberapa kemungkinan, seperti percobaan penarikan uang tanpa kecukupan ketersediaan dana, penarikan uang secara normal, penarikan dengan

penggunaan PIN yang salah, dan lainnya (Sholiq, 2006). Contoh diagram sekuensial disajikan pada Gambar 2.6 (John W. Satzinger, 2016).

Tabel 2. 3 Sequence Diagram

Notasi	Keterangan	Simbol
<i>Actor</i>	Pengguna sistem atau yang berinteraksi langsung dengan sistem, bisa manusia, aplikasi, ataupun objek lain.	
<i>Garis hidup /lifeline</i>	Menyatakan kehidupan suatu objek	
<i>Objek</i>	n objek yang berinteraksi dengan pesan	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama objek :nama kelas</div>
<i>Waktu aktif</i>	Objek dalam keadaan aktif dan berinteraksi.semuanya yang terhubung dengan waktu aktif	
<i>Pesan tipe create</i>	Suatu objek yang membuat objek yang lain	<< create >> 
<i>Pesan tipe call</i>	Suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri	1. nama_metode() 
<i>Pesan tipe send</i>	Suatu objek mengirimkan data ke objek lainnya	1.masukan 
<i>Pesan tipe return</i>	Suatu objek yang telah menjalankan suatu operasi, metode yang menghasilkan kembalian ke objek tertentu.	1.keluaran 

<i>Pesan tipe</i> <i>destroy</i>	Suatu objek mengakhiri hidup objek yang lain.	
---	---	--

2.10 Selection Sort

Algoritma pengurutan sederhana salah satunya adalah Selection Sort. Ide dasarnya adalah melakukan beberapa kali pass untuk melakukan penyeleksian elemen struktur data. Untuk sorting ascending (menaik), elemen yang paling kecil di antara elemen-elemen yang belum urut, disimpan indeksnya, kemudian dilakukan pertukaran nilai elemen dengan indeks yang disimpan tersebut dengan elemen yang paling depan yang belum urut. Sebaliknya, untuk sorting descending (menurun), elemen yang paling besar yang disimpan indeksnya kemudian ditukar.

Selection Sort diakui karena kesederhanaan algoritmanya dan performanya lebih bagus daripada algoritma lain yang lebih rumit dalam situasi tertentu. Algoritma ini bekerja sebagai berikut:

1. Mencari nilai minimum (jika ascending) atau maksimum (jika descending) dalam sebuah list
2. Menukarkan nilai ini dengan elemen pertama list
3. Mengulangi langkah di atas untuk sisa list dengan dimulai pada posisi kedua.

Secara efisien kita membagi list menjadi dua bagian yaitu bagian yang sudah diurutkan, yang didapat dengan membangun dari kiri ke kanan dan dilakukan pada saat awal, dan bagian list yang elemennya akan diurutkan.

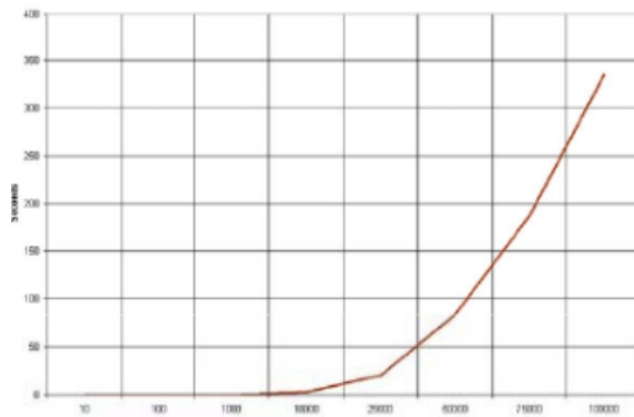
Algoritma di dalam Selection Sort terdiri dari kalang bersarang. Dimana kalang tingkat pertama (disebut pass) berlangsung $N-1$ kali. Di dalam kalang kedua, dicari elemen dengan nilai terkecil. Jika didapat, indeks yang didapat ditimpakan ke variabel min. Lalu dilakukan proses penukaran. Begitu seterusnya untuk setiap Pass. Pass sendiri

makin berkurang hingga nilainya menjadi semakin kecil. Berdasarkan operasi perbandingan elemennya:

$$T(n) = (n - 1) + (n - 2) + \dots + 2 + 1 = \sum_{i=1}^{n-1} n - i$$

$$= \frac{n(n-1)}{2} = O(n^2)$$

Berarti kompleksitasnya secara simptotik adalah $O(n^2)$. Adapun grafik efisiensi selection sort dapat dilihat pada tabel dibawah ini:



Gambar 2. 2 Grafik Kompleksitas Selection Sort

Selection sort merupakan perbaikan dari bubble sort dengan mengurangi jumlah perbandingan. Dalam selection sort, algoritma ini memilih satu persatu elemen data dari posisi awal, untuk mencari data paling kecil dengan mencatat posisi index nya saja. Adapun kelebihan yang dimiliki metode selection sort adalah sebagai berikut :

1. Algoritma selection sort mudah untuk di implementasikan.
2. Dapat mempercepat pencarian
3. Mudah menentukan data maksimum dan minimum.
4. Mudah digabungkan kembali.
5. Kompleksitas selection sort relatif lebih kecil.

2.11 System

Sistem berasal dari bahasa latin yaitu *systema* atau bahasa yunani *sustēma* yang berarti suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Sistem jugamerupakan sebuah kesatuan bagian-bagian yang saling memiliki hubungan yang berbeda dalam suatu wilayah, serta memiliki item-item sebagai penggerak.

Sistem didefinisikan menjadi “Suatu sistem adalah suatu jaringan kerja dari prosedurprosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sistem menurut Marshall B. Romney dan Paul Jhon Steinbart, Sistem (system) adalah serangkaian dua atau lebih komponen yang saling terkait dan berinteraksi untuk mencapai tujuan. Berdasarkan definisi diatas sistem adalah kumpulan atau serangkaian komponen yang saling terikat untuk mencapai suatu tujuan tertentu.(Kunci, 2020)

Menurut Jerry Fitzgerald dan Warren D. Stalling, Jr mengatakan bahwa “ sistem adalah kumpulan dari elemen-elemen yang berinteraksi serta jaringan kerja dari prosedur-prosedur yang saling berhubungan bersama-sama untuk melakukan sesuatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu” (Sudradjat, 2020). Menurut Kurnia Cahya Lestari dan Arni Muarifah Amri mengemukakan bahwa Sistem mengemukakan bahwa sistem adalah dua atau lebih komponen yang saling berhubungan dan berinteraksi membentuk kesatuan kelompok guna menghasilkan satu tujuan yang diharapkan. Sistem adalah sebuah tatanan yang mempunyai tujuan untuk memenuhi suatu proses tertentu karena sistem memiliki satuan dan tugas khusus yang saling berhubungan.(Dewi, 2021)

2.12 Penelitian Terkait

Adapun penelitian terkait pada penulisan yang peneliti lakukan adalah sebagai berikut : Lihat pada tabel dibawah :

Tabel 2. 4 Penelitian Terkait

No	Nama Penulis	Judul / Tahun Terbit	Keterangan
1	Yuni Arkiansyah	SISTEM REPOSITORY BERITA DAN GAMBAR PADA HARIAN UMUM LAMPUNG POST (2014)	Repository atau gudang sangat di butuhkan untuk kepentingan penyimpanan atau dokumentasi suatu naskah atau gambar pada PT. Masa Kini Mandiri (Lampung Post) khususnya di bagian dokumentasi redaksi, sehingga pada saat berita atau gambar yang di butuhkan suatu-waktu dapat di pergunakan kembali sebagaimana mestinya.
2	Raden Kania, Rustam Effendi, Muh. Rizal Sidik	SISTEM REPOSITORY PERPUSTAKAAN DI UNIVERSITAS BANTEN JAYA BERBASIS WEB (STUDI KASUS : UNIVERSITAS BANTEN JAYA) (2021)	Sistem ini memudahkan staf dalam pengelolaan karya ilmiah, dimana sistem tersebut dapat dengan mudah dipergunakan mahasiswa dalam mengupload hasil karya ilmiahnya. Selain itu staf dapat dengan mudah mencetak surat penyerahan laporan karya ilmiah tanpa harus membuka file excel yang biasa digunakan untuk mencetak ketika mahasiswa telah melakukan penyerahan skripsi.
3	Andi Christian, Muchlis	PENGEMBANGAN APLIKASI SISTEM INFORMASI	Dalam mengembangkan repository sistem informasi karya ilmiah menggunakan Metode

		<p>REPOSITORY KARYA ILMIAH PADA STMIK PRABUMULIH (2020)</p>	<p>Waterfall yang merupakan metode pengembangan desain sistem informasi secara sistematis melalui tahapan yang telah ditentukan. Diharapkan dengan pengembangan sistem informasi untuk mengelola repository karya ilmiah ini dapat menangani permasalahan pengelolaan karya ilmiah mahasiswa dan dosen sehingga dapat terlaksana secara efektif dan efisien.</p>
--	--	---	--