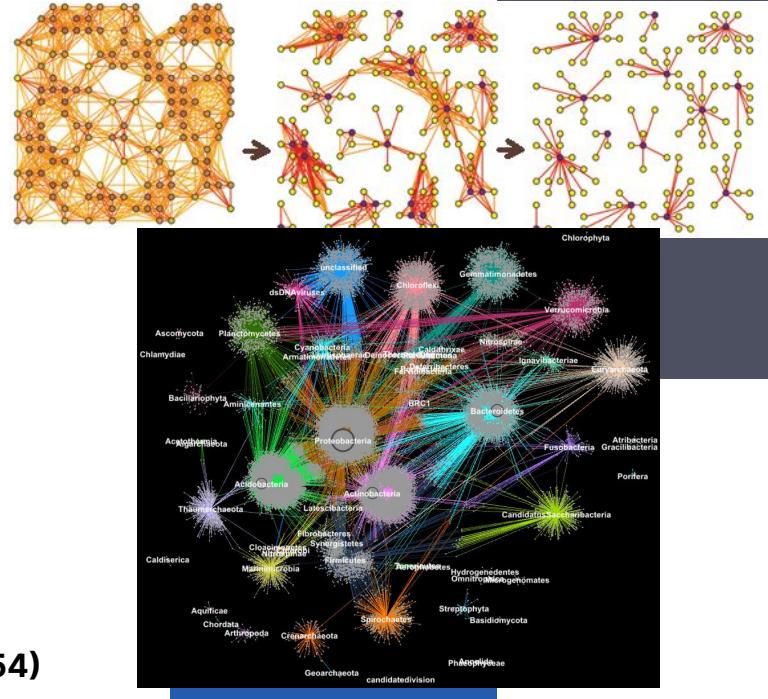




Tugas Proyek Akhir Kelompok

Kelompok Akatsuki

- Fakhri Perdana (1906352060)
- Richardy Lobo' Sapan (1906373954)



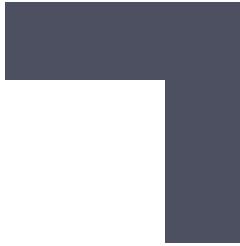
Latar Belakang

Dalam karya ini ini, penulis menjelaskan tentang parallel reduction dan Deep Learning dalam komputasi paralel. OpenMP pada dasarnya memiliki Klausula reduksi OpenMP, di mana Klausula reduksi OpenMP memungkinkan Anda menentukan satu atau lebih variabel thread-private yang tunduk pada operasi reduksi di akhir wilayah paralel. Deep Learning adalah jenis pembelajaran mesin dan kecerdasan buatan (AI) yang meniru cara manusia memperoleh jenis pengetahuan tertentu. Deep Learning adalah elemen penting dari ilmu data, yang mencakup statistik dan pemodelan prediktif.

Pada karya ini, penulis akan membahas tentang pengenalan singkat mengenai beberapa hal, diantaranya: (1) Bagaimana konsep dan proses parallel reduction serta cara implementasinya di OpenMP dan GPU Computing (CUDA), (2) Bagaimana contoh makalah tentang teknologi terbaru untuk GPU computing dan Deep learning untuk persoalan terkini bidang data science dan AI (Mis: Masalah Covid-19, big data, telemedicine, blockchain dll.) menggunakan platform terkini (Misal Rapids, KNIME, Torch, PyTorch, Keras, Tensorflow, Clara dll)?, dan (3) Bagaimana Contoh Implementasi GPU untuk aplikasi Deep Learning.

Parallel Reduction

L





2.1 Pembahasan konsep dan proses Paralel Reduction serta cara implementasinya di OpenMP dan GPU Computing (CUDA).



Konsep Parallel Reduction

Reduksi parallel adalah salah satu jenis algoritma PRAM. Reduksi parallel adalah proses memanipulasi data yang disimpan dalam register global. Reduksi paralel mengacu pada algoritma yang menggabungkan array elemen yang menghasilkan nilai tunggal sebagai hasilnya. Masalah yang memenuhi syarat untuk algoritma ini termasuk yang melibatkan operator yang bersifat asosiatif dan komutatif. Salah satu di antaranya termasuk penjumlahan array.

PRAM adalah mesin abstrak memori bersama. PRAM dimaksudkan sebagai analogi komputasi paralel dengan mesin akses acak (RAM). PRAM digunakan oleh perancang algoritma paralel untuk memodelkan kinerja algoritme paralel (seperti kompleksitas waktu, di mana jumlah prosesor diasumsikan biasanya juga dinyatakan).

Race Condition & Reduction

Race Condition terjadi ketika beberapa utas membaca dan menulis variabel secara bersamaan, misalnya:

```
asum = 0.0d0
 !$omp parallel do shared(x,y,n,asum) private(i)
  do i = 1, n
    asum = asum + x(i)*y(i)
  end do
 !$omp end parallel do
```

Hasil acak tergantung pada urutan asum akses utas. penulis membutuhkan beberapa mekanisme untuk mengontrol akses.

Race Condition & Reduction

Menjumlahkan elemen array adalah contoh operasi reduksi.

$$S = \sum_{j=1}^N A_j = \sum_{j=1}^{\frac{N}{2}} A_j + \sum_{\substack{j=N \\ j=\frac{N}{2}+1}}^N A_j = B_1 + B_2 = \sum_{j=1}^2 B_j$$

OpenMP menyediakan dukungan untuk reduksi umum dalam region paralel dan loop dengan klausula reduksi sebagai berikut.

Race Condition & Reduction

reduction(operator:list)

Melakukan pengurangan pada variabel (skalar) dalam daftar

- Variabel pengurangan pribadi dibuat untuk hasil parsial setiap utas
- Variabel reduksi pribadi diinisialisasi ke nilai awal operator
- Setelah wilayah paralel, operasi pengurangan diterapkan ke variabel pribadi dan hasilnya digabungkan ke variabel bersama

Berikut adalah operator reduksi yang ada dalam bahasa C/C++ dan Fortran.

Reduction operators in C/C++

Operator	Initial value
+	0
-	0
*	1
&&	1
	0

Bitwise Operator	Initial value
&	~ 0
	0
^	0

Reduction operators in Fortran

Operator	Initial value
+	0
-	0
*	1
max	least
min	largest
.and.	.true.
.or.	.false.
.eqv.	.true.
.neqv.	.false.

Bitwise Operator	Initial value
.iand.	all bits on
.ior.	0
.ieor.	0

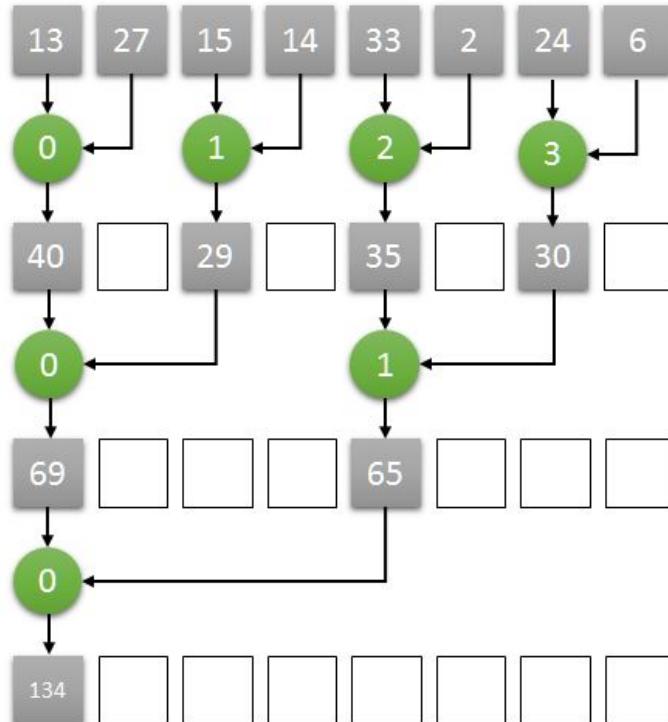


Race condition avoided with reduction clause

```
!$omp parallel do shared(x,y,n) private(i) reduction(+:asum)
do i = 1, n
    asum = asum + x(i)*y(i)
end do
 !$omp end parallel do
```

```
#pragma omp parallel for shared(x,y,n) private(i) reduction(+:asum)
for(i=0; i < n; i++) {
    asum = asum + x[i] * y[i];
}
```

Proses Parallel Reduction dengan CUDA



Berikut adalah ide utamanya:

- Dengan asumsi N sebagai jumlah elemen dalam array, kita memulai $N/2$ utas, satu utas untuk setiap dua elemen
- Setiap utas menghitung jumlah dari dua elemen yang sesuai, menyimpan hasilnya pada posisi yang pertama.
- Secara iteratif, setiap langkah:
 - jumlah utas dibelah dua (misalnya, dimulai dengan 4, lalu 2, lalu 1)
 - menggandakan ukuran langkah antara dua elemen yang sesuai (dimulai dengan 1, lalu 2, lalu 4)
- Setelah beberapa iterasi, hasil akhir dari reduksi akan disimpan di elemen pertama array

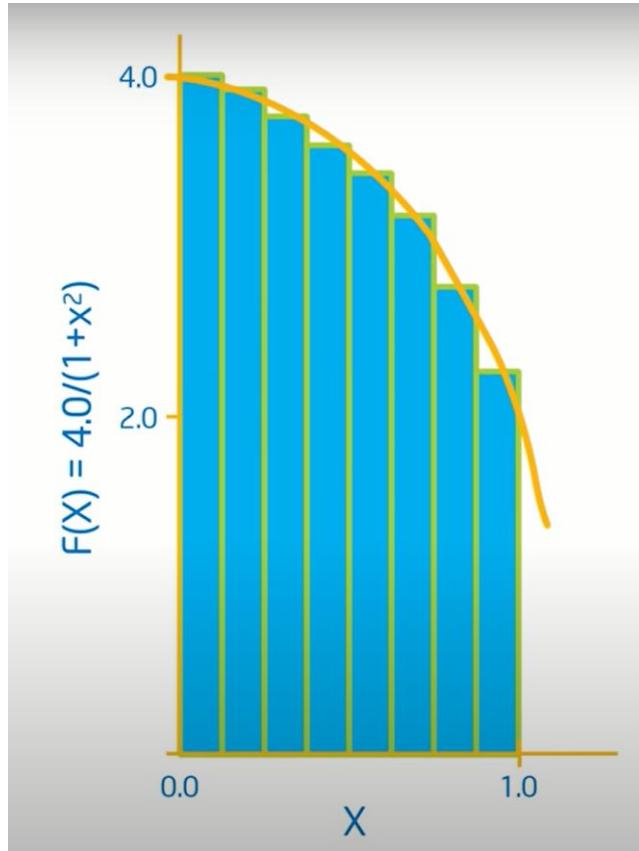


Pi Calculation



Simulasi OpenMP: Menghitung Pi (integral)

$$\int_0^1 \frac{4.0}{(1+x^2)} dx = \pi$$



Simulasi OpenMP: Menghitung Pi (integral) - Sekuensial

```
10 #include <stdio.h>
11 #include <omp.h>
12 static long num_steps = 100000000;
13 double step;
14 int main ()
15 {
16     int i;
17     double x, pi, sum = 0.0;
18     double start_time, run_time;
19
20     step = 1.0/(double) num_steps;
21
22
23     start_time = omp_get_wtime();
24
25     for (i=1;i<= num_steps; i++){
26         x = (i-0.5)*step;
27         sum = sum + 4.0/(1.0+x*x);
28     }
29
30     pi = step * sum;
31     run_time = omp_get_wtime() - start_time;
32     printf("\n pi with %ld steps is %lf in %lf seconds\n ",num_steps,pi,run_time);
33 }
```

Program ini menghitung integral dari fungsi

$$4/(1+x^2)$$

dari 0 ke 1 secara numerik. Nilai integralnya = pi.

Program ini merupakan versi sequential.
Program menggunakan timer dari openmp

```
C:\Users\A\Desktop\Serial pi calc\bin\Debug\parallel pi calc.exe

pi with 100000000 steps is 3.141593 in 0.463000 seconds
Process returned 0 (0x0)  execution time : 0.513 s
Press any key to continue.
```

Simulasi OpenMP: Menghitung Pi (integral) - Paralel

```
18 #include <stdio.h>
19 #include <omp.h>
20 static long num_steps = 100000000;
21 double step;
22 int main ()
23 {
24     int i;
25     double x, pi, sum = 0.0;
26     double start_time, run_time;
27
28     step = 1.0/(double) num_steps;
29     for (i=1;i<=4;i++) {
30         sum = 0.0;
31         omp_set_num_threads(i);
32         start_time = omp_get_wtime();
33 #pragma omp parallel
34 {
35 #pragma omp single
36     printf(" num_threads = %d",omp_get_num_threads());
37
38 #pragma omp for reduction(+:sum)
39     for (i=1;i<= num_steps; i++) {
40         x = (i-0.5)*step;
41         sum = sum + 4.0/(1.0+x*x);
42     }
43 }
44     pi = step * sum;
45     run_time = omp_get_wtime() - start_time;
46     printf("\n pi is %f in %f seconds and %d threads\n",pi,run_time,i);
47 }
48 }
```

Program ini akan menghitung nilai fungsi $4/(1+x^2)$ dari 0 ke 1 secara numerik. Nilai integral ini = pi. Program diparalelisasi dengan menambahkan 4 baris,

- (1) Baris yg mengandung omp.h -- file include yg mengandung OpenMP's function prototypes dan constants.
- (2) Pragma utk membuat sekelompok threads
- (3) Pragma yang menyebabkan salah satu thread mencetak jumlah thread yang digunakan oleh program.
- (4) Pragma untuk membagi iterasi loop di antara kelompok threads. Pragma ini mencakup 2 klausa untuk (1) membuat variabel pribadi dan (2) menyebabkan thread menghitung jumlahnya secara lokal dan kemudian menggabungkan jumlah lokalnya menjadi satu nilai global.

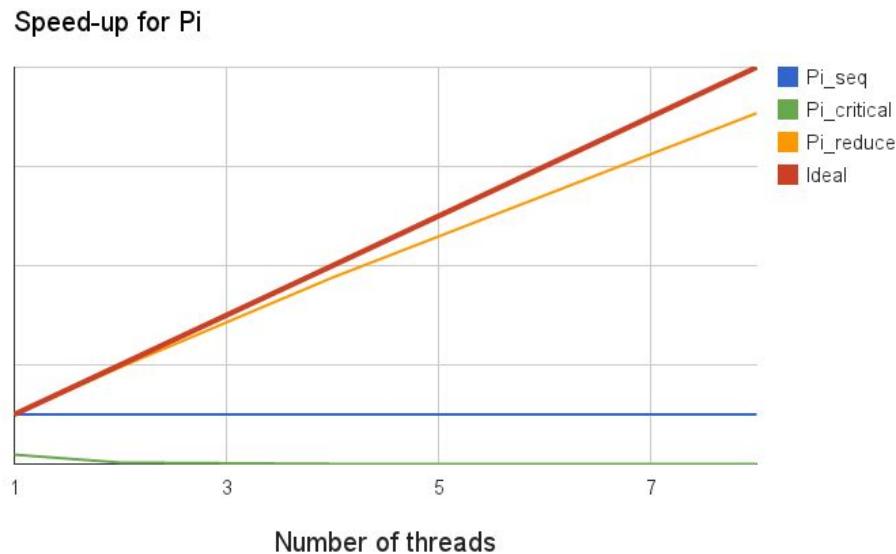
Simulasi OpenMP: Menghitung Pi (integral) - Paralel

```
"C:\Users\A\Desktop\serial pi calc\bin\Debug\serial pi calc.exe"
```

```
num_threads = 1
pi is 3.141593 in 0.415000 seconds and 1 threads
num_threads = 2
pi is 3.153437 in 0.839000 seconds and 2 threads
num_threads = 3
pi is 3.140003 in 0.761000 seconds and 3 threads
num_threads = 4
pi is 3.140808 in 0.671000 seconds and 4 threads
```

```
Process returned 0 (0x0)    execution time : 2.728 s
Press any key to continue.
```

Perbandingan



Speed-up yang ideal adalah jumlah thread, misalnya jika aplikasi bekerja dengan 4 thread, speed-up yang ideal dalam hal ini adalah 4.

Paralelisasi dengan klausa reduksi hampir mencapai speed up ideal, sebaliknya dengan speed up sekuensial lebih buruk. Tampak jelas bahwa dengan paralelisasi, efektivitas dan produktivitas dapat lebih meningkat dibandingkan program sekuensial. Selain cara di atas, parallel reduction pada OpenMP juga dapat dilakukan dengan cara sebagai berikut.

Approximation of PI hand-crafting the #pragma omp reduction

```
h = 1.0 / n;

#pragma omp parallel private(x) shared(n, h)
{
    double thread_area = 0;                                // Private / local variable

    #pragma omp for
    for (i = 1; i <= n; i++)
    {
        x = h * (i - 0.5);
        thread_area += (4.0 / (1.0 + x*x));
    }

    #pragma omp atomic
    area += thread_area;                                    // Applies the reduction manually
                                                        // All threads aggregate into area
}

pi = h * area;
    return 0;
}
```

Utas muncul dalam paralel #pragma omp. Setiap utas akan memiliki thread_area independen/pribadi yang menyimpan sebagian tambahannya. Loop berikut didistribusikan di antara utas menggunakan #pragma omp for. Dalam loop ini, setiap thread menghitung thread_areanya sendiri dan setelah loop ini, kode secara berurutan menggabungkan area secara atom melalui #pragma omp atomic.

Approximation of PI using #pragma omp reduction clause

```
h = 1.0 / n;
#pragma omp parallel for private(x) shared(n, h) reduction(+:area)
for (i = 1; i <= n; i++)
{
    x = h * (i - 0.5);
    area += (4.0 / (1.0 + x*x));
}
pi = h * area;
```

Dalam contoh ini, setiap utas mengeksekusi subset dari jumlah iterasi. Setiap utas memiliki salinan area pribadi lokalnya dan pada akhir wilayah paralel mereka semua menerapkan operasi penambahan (+) sehingga menghasilkan nilai akhir untuk area.

Approximation of PI using reductions based on #pragma atomic

```
int main()
{
    h = 1.0 / n;
    #pragma omp parallel for private(x) shared(n, h, area)
    for (i = 1; i <= n; i++)
    {
        x = h * (i - 0.5);
        #pragma atomic
        area += (4.0 / (1.0 + x*x));
    }
    pi = h * area;

    return 0;
}
```

Dalam contoh ini, setiap utas mengeksekusi subset dari jumlah iterasi dan mereka terakumulasi secara atom ke dalam area variabel bersama, yang memastikan bahwa tidak ada pembaruan yang hilang. Kita dapat menggunakan #pragma atomik di sini karena operasi yang diberikan ($+=$) dapat dilakukan secara atom, yang menyederhanakan keterbacaan dibandingkan dengan penggunaan #pragma omp critical

Approximation of PI using reductions based on #pragma omp critical

```
h = 1.0 / n;
#pragma omp parallel for private(x) shared(n, h, area)
for (i = 1; i <= n; i++)
{
    x = h * (i - 0.5);
    #pragma omp critical
    {
        area += (4.0 / (1.0 + x*x));
    }
}
pi = h * area;
```

Dalam contoh ini, setiap utas mengeksekusi subset dari jumlah iterasi dan mereka terakumulasi secara atomik dalam area variabel bersama, yang memastikan bahwa tidak ada pembaruan yang hilang.

Implementasi Menggunakan CUDA



```
%%cu
#include "cuda_runtime.h"
#include "device_launch_parameters.h"

#include <iostream>
#include <numeric>
using namespace std;

__global__ void sum(int* input)
{
    const int tid = threadIdx.x;

    auto step_size = 1;
    int number_of_threads = blockDim.x;

    while (number_of_threads > 0)
    {
        if (tid < number_of_threads) // still alive?
        {
            const auto fst = tid * step_size * 2;
            const auto snd = fst + step_size;
            input[fst] += input[snd];
        }
    }
}
```

```
    step_size <= 1;
    number_of_threads >= 1;
}
}

int main()
{
    const auto count = 8;
    const int size = count * sizeof(int);
    int h[] = {13, 27, 15, 14, 33, 2, 24, 6};

    int* d;

    cudaMalloc(&d, size);
    cudaMemcpy(d, h, size, cudaMemcpyHostToDevice);

    sum <<<1, count / 2 >>>(d);

    int result;
    cudaMemcpy(&result, d, sizeof(int), cudaMemcpyDeviceToHost);

    cout << "Sum is " << result << endl;

    getchar();
}
```

```
cudaFree(d);
delete[] h;

return 0;
```

Sum is 134



2.2 Makalah tentang teknologi terbaru untuk GPU computing dan Deep learning untuk persoalan terkini bidang data science dan AI (Mis: Masalah Covid-19, big data, telemedicine, blockchain dll.) menggunakan platform terkini (Misal Rapids, KNIME, Torch, PyTorch, Keras, Tensorflow, Clara dll).



Makalah Deep Learning

Sendak M, Ratliff W, Sarro D, Alderton E, Futoma J, Gao M, Nichols M, Revoir M, Yashar F, Miller C, Kester K, Sandhu S, Corey K, Brajer N, Tan C, Lin A, Brown T, Engelbosch S, Anstrom K, Elish M, Heller K, Donohoe R, Theiling J, Poon E, Balu S, Bedoya A, O'Brien C

Real-World Integration of a Sepsis Deep Learning Technology Into Routine Clinical Care: Implementation Study

JMIR Med Inform 2020;8(7):e15182

URL: <https://medinform.jmir.org/2020/7/e15182>

DOI: 10.2196/15182

Latar Belakang

Pembelajaran mesin telah dengan cepat diadopsi dalam ilmu biomedis untuk meningkatkan metode prediktif, prognostik, dan diagnostik. Namun, banyak hambatan teknis dan klinis untuk adopsi tetap ada. Pertama, catatan kesehatan elektronik (EHR) seringkali tidak memiliki fungsi asli untuk mengintegrasikan model pembelajaran mesin yang kompleks. Diperlukan investasi yang signifikan dalam infrastruktur. Kedua, bahkan setelah model awalnya diimplementasikan, model pembelajaran mesin dapat menimbulkan biaya pemeliharaan berkelanjutan yang substansial. Ketiga, meskipun beberapa sistem kesehatan membangun dan mengintegrasikan solusi pembelajaran mesin yang dikembangkan sendiri, upaya itu sering dialihdayakan ke tim peneliti atau vendor teknologi. Pemisahan antara operasi dan implementasi serta pemeliharaan model ini menghadirkan tantangan tambahan, karena "kepemilikan rekayasa dari sinyal input terpisah dari kepemilikan rekayasa model yang mengkonsumsinya." Akhirnya, banyak model tidak terintegrasi secara efektif ke dalam alur kerja klinis dengan cara yang meningkatkan perawatan klinis atau hasil.

Sepsis Watch

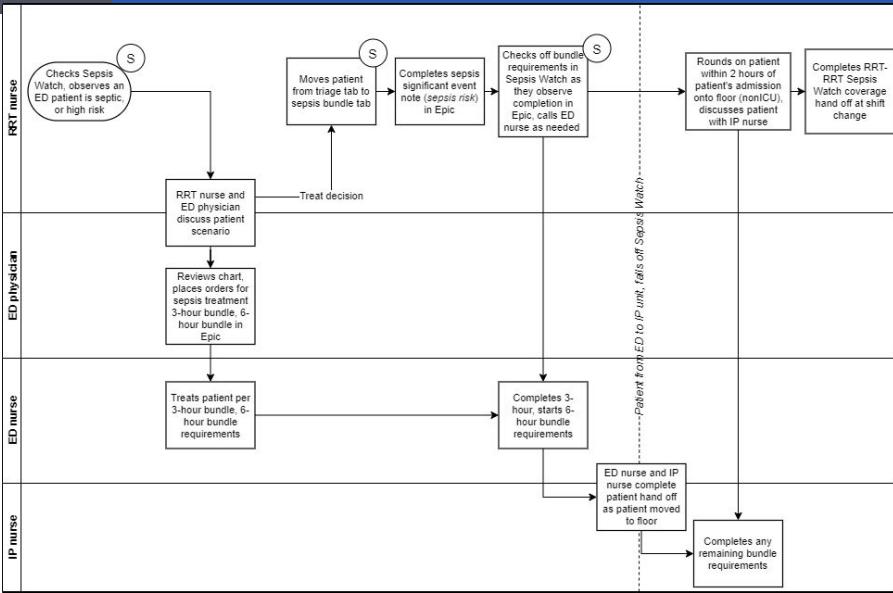
Di sini, penulis mempelajari detail tentang bagaimana sistem kesehatan mengintegrasikan teknologi pembelajaran mendalam skala penuh pertama ke dalam perawatan klinis rutin. Sebuah kelompok inovasi menghabiskan lebih dari dua tahun dengan mitra di seluruh organisasi untuk meluncurkan solusi pembelajaran mendalam, Sepsis Watch, pada 5 November 2018. Sepsis Watch adalah platform deteksi dan manajemen sepsis yang digunakan oleh dokter untuk meningkatkan kepatuhan terhadap pedoman pengobatan yang direkomendasikan untuk sepsis dan sehingga meningkatkan hasil pasien. Meskipun Sepsis Watch adalah contoh dukungan keputusan klinis pembelajaran mesin (CDS), sistem pembelajaran mendalam memang menimbulkan tantangan implementasi di luar CDS tradisional, seperti yang dirinci di tempat lain. Secara khusus, mekanisme kepercayaan dan akuntabilitas baru harus dikembangkan untuk memastikan bahwa sistem tersebut aman dan dapat diandalka.

Sepsis Watch

Pada Tabel berikut, penulis menyajikan 8 langkah yang diperlukan untuk mengintegrasikan Sepsis Watch ke dalam pemberian perawatan rutin dengan sukses. Penulis mengambil pelajaran dari kerangka kerja sistem kesehatan pembelajaran dan praktik terbaik yang dijelaskan sebelumnya untuk pembelajaran mesin yang bertanggung jawab dalam perawatan kesehatan. Tujuan dari naskah ini adalah untuk menggambarkan setiap langkah secara rinci dan menyoroti pembelajaran yang dapat menginformasikan upaya terkait di organisasi lain.

Tabel 1: Langkah Implementasi
Machine Learning pada Sepsis Watch

Sepsis Watch



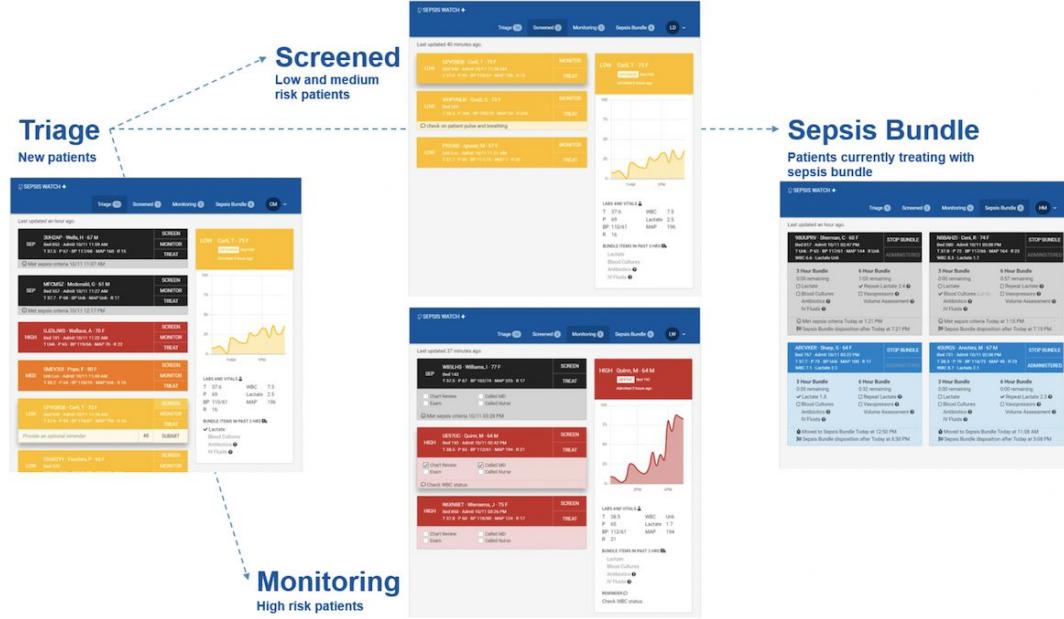
Gambar 1: Swimlane Diagram Sepsis
Watch

Table 1. Steps for integrating machine learning into clinical care. The table includes definitions for the various steps and example tasks and deliverables during the step.

Step in the process	Definition	Example tasks and milestones
• Problem assessment	<ul style="list-style-type: none"> Understand the root cause of the problem, the magnitude of the problem, where the problem is felt most acutely, who is best positioned to address the problem, and what changes need to occur to empower someone to address the problem 	<ul style="list-style-type: none"> Data analysis to understand the magnitude, setting, and timing of the problem Observe frontline staff in clinical settings where the problem occurs Interview a broad group of stakeholders to understand complexities in addressing the problem
• Internal and external scans of solutions and workflows	<ul style="list-style-type: none"> Perform due diligence on internal and external tools that attempt to address the problem 	<ul style="list-style-type: none"> Evaluate technologies and workflows available through current information technology supplier relationships Evaluate technologies on the market sold by external vendors Interview internal stakeholders who have previously attempted to solve the problem
• Clinical workflow design	<ul style="list-style-type: none"> Design clinical workflow that integrates new technology to address the problem 	<ul style="list-style-type: none"> Gather requirements from frontline staff and leadership Iterate on workflow designs with frontline staff Identify constraints (eg, time and effort) to ensure that the end user is able to use the technology effectively
• Model and infrastructure design	<ul style="list-style-type: none"> Design machine learning model and accompanying infrastructure to ensure that the technology can effectively be integrated into clinical workflows 	<ul style="list-style-type: none"> Identify a set of input features used by the model to address the problem, making sure to incorporate clinical domain expertise and prior literature Design infrastructure to support clinical decisions in a timely, actionable manner Identify performance metrics and goals that are most important and relevant to stakeholders and end-users
• Clinical workflow application development	<ul style="list-style-type: none"> Develop the clinical workflow application and integrations with other technologies 	<ul style="list-style-type: none"> Develop user interface and user experience Integrate with electronic health record to access the required data at the required latency Prototype workflow application with end users
• Model and infrastructure development	<ul style="list-style-type: none"> Develop the machine learning model and infrastructure required to implement model, including integrations with other technologies 	<ul style="list-style-type: none"> Develop and validate the machine learning model on retrospective data Validate the machine learning model and infrastructure on prospective <i>silent period</i> launch
• Implementation, change management, and governance	<ul style="list-style-type: none"> Implement the machine learning model with accompanying education, communication, and governance to ensure accountability and successful adoption 	<ul style="list-style-type: none"> Establish a governance committee with agreed-upon tasks and mission Develop training material to ensure end users effectively use the new technology Communicate broadly about the technology implementation and roles and responsibilities
• Evaluation plan and partnerships	<ul style="list-style-type: none"> Prespecify evaluation plan, target goals, and safety and efficacy monitoring 	<ul style="list-style-type: none"> Develop internal and external partnerships to ensure rigorous evaluation Register clinical trial

Sepsis Watch

Gambar 2: Design User Interface Sepsis Watch



Sepsis Watch

The Problem: Struggle with sepsis

1 What is the problem? Sepsis

Location	Percent of patients who received appropriate care
Duke University Hospital	22%
Duke Regional Hospital	40%
Duke Raleigh Hospital	23%
North Carolina average	50%
National average	49%

¹CMS 2015 sepsis data

2 Where does the problem occur? The ED

Over a 14-month timeframe, we found that the majority of sepsis diagnoses occurred in the ED. The dotted outline shows the sepsis cases diagnosed over a 10 hour window: 5 hours before and after admission orders are placed ("Hour 0")

3 Why are we failing to solve the problem today? Slow, false alarms

(fake) NEWS

- Only 6.8% of patients with NEWS (National Early Warning Score) BPA had discharge diagnosis of sepsis
- BPA fired 447 times/day on 42 unique patients/day on average (up to ~100x/patient).
- 63% of BPAs canceled.

The Solution: Sepsis Watch

1 Define adult sepsis at Duke

Criteria	Definition
2 or more SIRS criteria	- Temperature $>38^{\circ}\text{C}$ or $<36^{\circ}\text{C}$ (6 hours) - HR >90 (6 hours) - RR >20 (6 hours) - WBC count >12 , <4 , or % bandemia $>10\%$ (24 hours)
Suspect infection	- Blood culture order (24 hours)
1 element of end organ failure	- Creatinine >2.0 (24 hours) - INR >1.5 (24 hours) - Total bilirubin >2.0 (24 hours) - SBP <90 or decrease in SBP by >40 (6 hours) - Platelets <100 (24 hours) - Lactate ≥ 2 (24 hours)

2 Create machine learning model to predict sepsis quickly and accurately

- 42,000+ inpatient encounters analyzed at Duke Hospital over 14 months, 21.3% with a sepsis event.
- 32+ million data points incorporated: 25 million vital sign measurements, 2 million med admins, 2.5 million labs.
- 34 physiological variables (5 vitals, 29 labs).
- At least one value for each vital in 99% of encounters.
- Some labs rarely measured (2-4%), most measured 20-80% of the time.
- 35 baseline covariates (e.g. age, transfer status, comorbidities).
- 10 medication classes (antibiotics, opioids, heparins).

3 Design web application to show real-time model results and track treatment

Sepsis Watch is a custom-built tool to support identification of patients at risk of sepsis in the hospital. Access Sepsis Watch App at <https://sepsiswatch.duke.edu>

Gambar 3: Sepsis Watch training
one-page overview.

Sepsis Watch

Metric types ^a	Metrics
Volume	Average number of new patients appearing on the Sepsis Watch Triage tab per day
Volume	Distribution of new patients appearing on the Sepsis Watch Triage tab, by hour of the day
Volume	Median length of time patient remained on the Sepsis Watch Triage tab before being moved to another tab
Volume	Average number of patients moved to the Sepsis Bundle Treatment tab per day
Bundle compliance	3-hour bundle compliance for patients moved to the Sepsis Watch Treatment tab (comprised of antibiotics, lactate, and blood culture 3-hour bundle components). Includes week-by-week performance
Bundle compliance	Antibiotics administration 3-hour compliance for patients moved to the Sepsis Watch Treatment tab. Includes week-by-week performance
Bundle compliance	Serum lactate collected 3-hour compliance for patients moved to the Sepsis Watch Treatment tab. Includes week-by-week performance
Bundle compliance	Blood culture collected 3-hour compliance for patients moved to the Sepsis Watch Treatment tab. Includes week-by-week performance

^aMetrics were chosen by the Sepsis Watch governance committee to present data for 2 distinct patient cohorts: (1) patients who met Sepsis Watch sepsis criteria and (2) patients who were at high risk for meeting Sepsis Watch sepsis criteria as identified by the model.

Tabel 2: Metrik laporan mingguan tata kelola
Sepsis Watch.

Sepsis Watch

April 2016 – month 0

November 2018 – month 32

Phase 1: months 0 to 6

Problem Assessment

- Key Stakeholders: frontline clinical staff, inpatient quality leadership, information technology leadership, innovation management
- Functions: identify data sources, curate data, interview staff and stakeholders, observe workflows
- Resources: personnel time, compute environment

Internal and External Scans of Solutions and Workflows

- Key Stakeholders: frontline clinical staff, inpatient quality leadership, information technology leadership, innovation management
- Functions: literature review, due diligence on vendor solutions and electronic health record clinical decision support
- Resources: personnel time

Phase 2: months 4 to 30

Clinical Workflow Design

- Key Stakeholders: frontline clinical staff, user interface design, user experience design, innovation management, information technology leadership
- Functions: develop and iterate on prototypes, design minimal viable product, interview staff and stakeholders, observe workflows
- Resources: personnel time

Model and Infrastructure Design

- Key Stakeholders: machine learning experts, data scientists, data engineers, innovation management, information technology leadership, frontline clinical staff
- Functions: gather requirements for model and infrastructure, identify data sources and required technology integrations
- Resources: personnel time

Clinical Workflow Application Development

- Key Stakeholders: user interface design, user experience design, innovation management, frontline clinical staff
- Functions: develop user interface
- Resources: personnel time

Model and Infrastructure Development

- Key Stakeholders: machine learning experts, data scientists, data engineers, innovation management, information technology leadership, frontline clinical staff
- Functions: develop technology integrations, develop data pipeline and data extraction, transform, and load system, develop and validate machine learning model
- Resources: personnel time, compute environment

Phase 3: months 24 to 32

Implement, Change Management, and Governance

- Key Stakeholders: frontline clinical staff, inpatient quality leadership, hospital leadership, emergency department leadership, information technology leadership, nursing leadership, innovation management
- Functions: relationship building, communication, training of frontline clinical staff
- Resources: personnel time, compute environment

Evaluation Plan and Partnerships

- Key Stakeholders: internal and external evaluation partners, frontline clinical staff, innovation management
- Functions: navigate regulatory and ethics review board processes, develop partnerships, design and register clinical trial
- Resources: personnel time

Gambar 4: Garis waktu

langkah-langkah yang terlibat dalam terjemahan Sepsis Watch dari identifikasi masalah hingga integrasi ke dalam perawatan klinis rutin.

Sepsis Watch

Kesimpulan

Terlepas dari keterbatasannya, keberhasilan integrasi Sepsis Watch ke dalam perawatan klinis rutin menandakan perjalanan melintasi jurang bagi Duke Health. Awalnya, sejumlah kecil dokter dan administrator visioner sangat ingin menggunakan teknologi yang muncul untuk mengatasi masalah klinis yang penting. Seiring perkembangan proyek selama dua tahun, kelompok pemangku kepentingan yang lebih luas menjadi sadar akan dampak potensial dari mengintegrasikan pembelajaran mesin ke dalam perawatan klinis. Permintaan baru untuk aplikasi diumumkan sebulan sebelum Sepsis Watch diluncurkan pada November 2018, dan Duke Institute for Health Innovation menerima rekor jumlah proposal pembelajaran mesin, di mana lima proposal pembelajaran mesin akhirnya dipilih oleh kepemimpinan senior dan diluncurkan pada bulan April 2019. Pada Juni 2019, Sepsis Watch disosialisasikan ke UGD di dua rumah sakit komunitas Duke Health. Banyak tantangan yang dihadapi selama proses integrasi, tetapi fokus pada peningkatan perawatan pasien memindahkan Sepsis Watch dari konsep ke desain ke produksi. Tidak ada pedoman tentang cara mengintegrasikan pembelajaran mesin ke dalam perawatan klinis, dan masih banyak lagi implementasi yang berhasil diperlukan untuk mengembangkan praktik terbaik. Pembelajaran dari integrasi Sepsis Watch telah menginformasikan proses yang dirancang untuk meningkatkan pelaksanaan proyek pembelajaran mesin dalam sistem kesehatan kita. Pembelajaran ini dapat memberikan arahan kepada tim yang mengejar integrasi pembelajaran mesin ke dalam perawatan di tempat lain.

2.3 Contoh Implementasi GPU untuk aplikasi Deep Learning: AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit



AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

Link

<https://github.com/richardy-lobo-sapan/AI-for-Healthcare-Project-using-NVIDIA-Jetson-Nano-2GB-Developer-kit>

Proyek ini menggunakan konsep Deep learning dalam mendeteksi Berbagai Penyakit Mematikan. Dapat Mendeteksi 1) Kanker Paru 2) Covid-19 3) Tuberkulosis 4) Pneumonia. Menggunakan CT-Scan dan X-ray Gambar dada/paru-paru dalam mendeteksi penyakit. Ini memiliki Akurasi antara 50% -80%. Itu dapat mengambil input dalam format Gambar apa pun atau melalui video Langsung dan memberikan output yang akurat.

Github:

The screenshot shows a GitHub repository page. At the top, there's a header with navigation links like 'Code', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, the repository name 'richardy-lobo-sapan / AI-for-Healthcare-Project-using-NVIDIA-Jetson-Nano-2GB-Developer-kit' is displayed, along with a 'Public' badge and options to 'Pin', 'Watch', 'Fork', and 'Star'. The main content area shows a list of files and their commit history. A summary at the top of the list states: 'This branch is up to date with THEGURU11/AI-for-Healthcare-Project-using-NVIDIA-Jetson-Nano-2GB-Developer-kitmain.' Below this, a list of commits is shown, each with a timestamp and a brief description. The commits are as follows:

Commit	Description	Date	Author
4ed1422	Update Building Project from scratch.md	on Dec 25, 2021	THEGURU11
Building Project from scratch.md	Update Building Project from scratch.md	6 months ago	
DATA & Models.pdf	Add files via upload	6 months ago	
LICENSE	Initial commit	6 months ago	
README.md	Update README.md	6 months ago	
Report file.pdf	Add files via upload	6 months ago	
Running Pre-Build Project.md	Update Running Pre-Build Project.md	6 months ago	

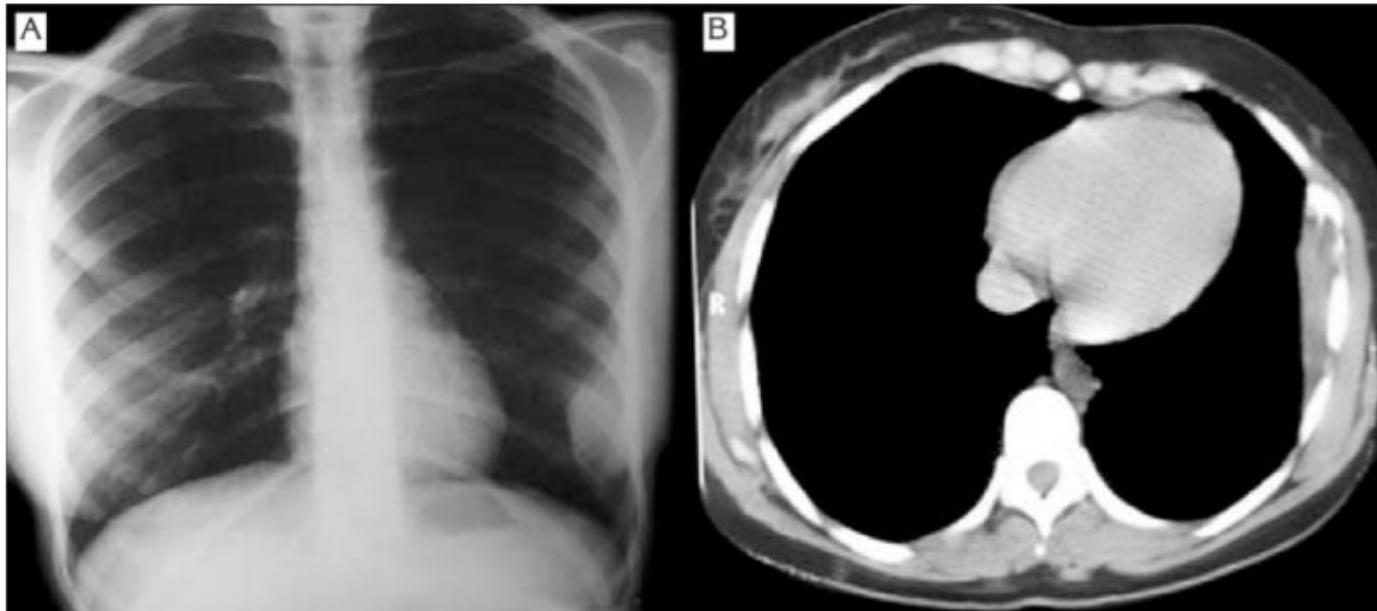
At the bottom of the page, there's a status bar showing the file 'AI-for-Healthcare-....zip' and the Windows taskbar with various application icons. The system tray indicates the temperature is 21°C and it's raining.

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

Proyek ini menggunakan konsep Deep learning dalam mendeteksi Berbagai Penyakit Mematikan.

- Dapat Mendeteksi 1) Kanker Paru 2) Covid-19 3) Tuberkulosis 4) Pneumonia.
- Menggunakan CT-Scan dan X-ray Gambar dada/paru-paru dalam mendeteksi penyakit.
- Model ini memiliki Akurasi antara 50% -80%.
- Model ini dapat mengambil input dalam format Gambar apa pun atau melalui video Langsung dan memberikan hasil keluaran yang akurat.

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit



| X-ray Image of chest |

| CT-Scan Image of Chest |

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

Atribusi

Menggunakan inferensi Jetson @ dusty-nv
(<https://github.com/dusty-nv/jetson-inference>)

Menggunakan Datasets:

1. CT-Scan :
<https://www.kaggle.com/mohamedhanyyy/che st-ctscan-images/download>
2. Rontgen :
<https://www.kaggle.com/jtiptj/che st-xray-pneumoniacovid19tuberculosis/download>

Datasets:

- CT-Scan:-<https://www.kaggle.com/mohamedhanyyy/che st-ctscan-images/download>
- X-ray:-<https://www.kaggle.com/jtiptj/che st-xray-pneumoniacovid19tuberculosis/download>

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

Accessories & Resources

- Jetson Developer Kit (2gb kit)
- Type C power (5V) supply
- Ethernet cable
- HDMI Cable
- Monitor with HDMI cable
- Camera (Logitech C270 HD WEBCAM)
- Keyboard & Mouse (wireless)
- Memory card (more than 32 GB)
- Optional: cooling fan, micro-USB cable(for headless mode)
- Jetson-Inference With Docker File: <https://github.com/dusty-nv/jetson-inference>

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

Akurasi Model:



AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

Langkah =>



1] Mengumpulkan semua Aksesoris:-

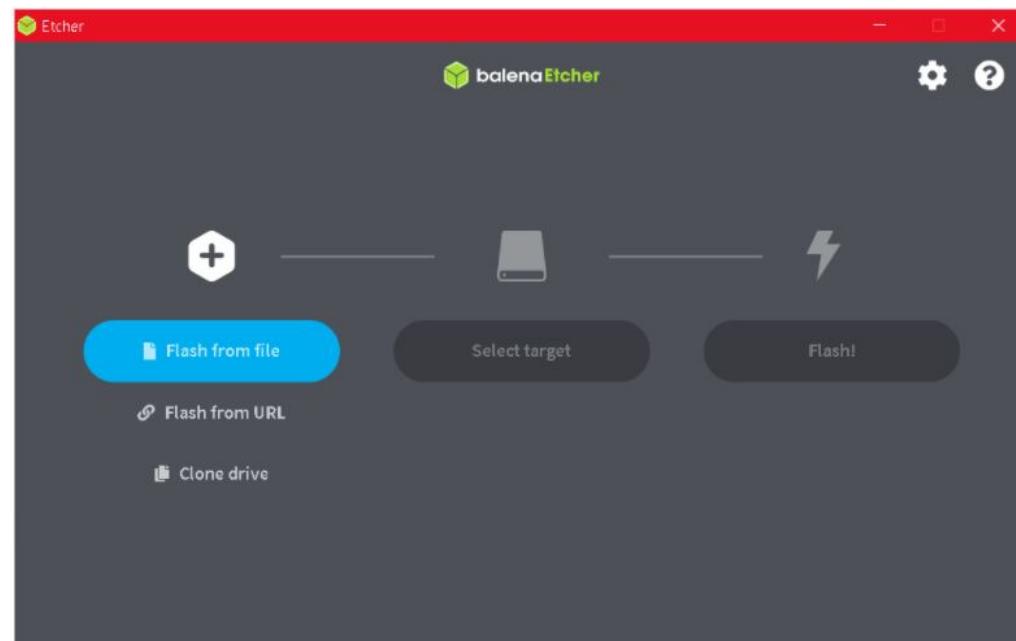
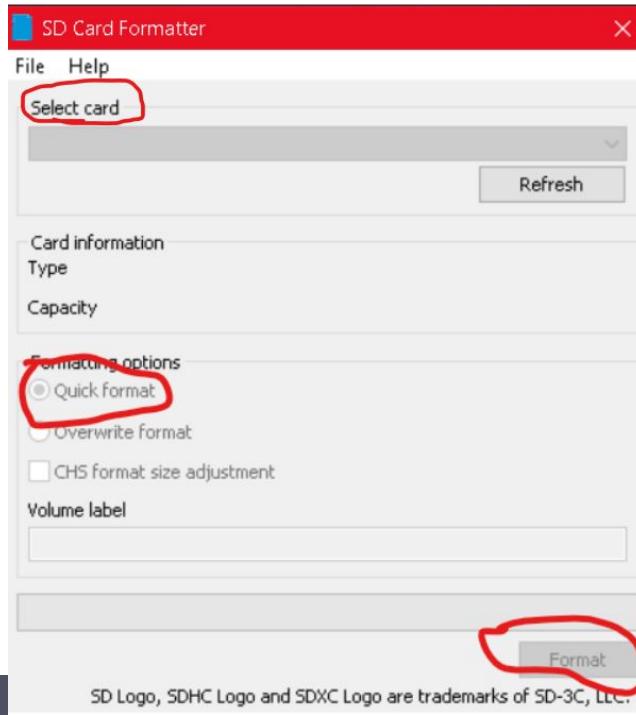
Kita mengumpulkan semua Aksesoris yang disebutkan di atas dari pasar lokal dan mengumpulkan 2GB Developer kit.

2] Mempersiapkan Penyiapan: -

- Hubungkan kartu SD ke PC/Laptop
- Unduh SD Card Image (Untuk kit 2 GB)
- Unduh SD Card Formatter & Instal.

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

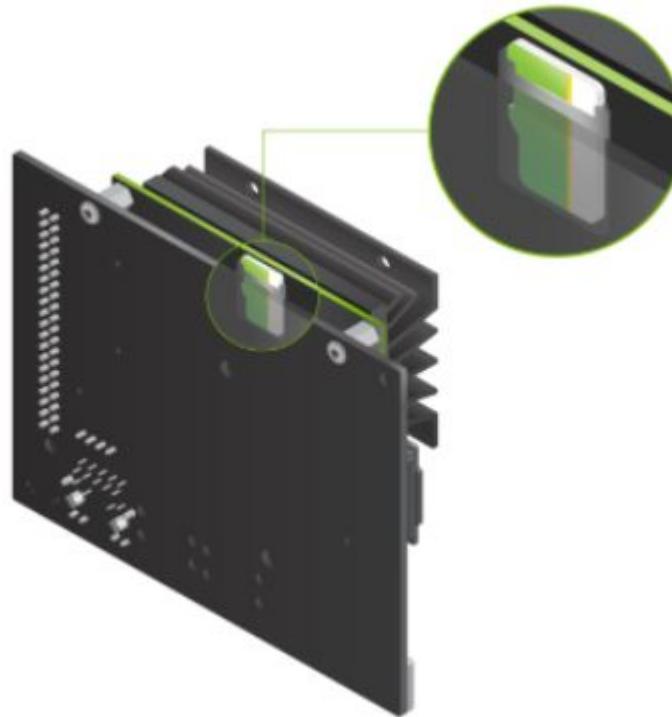
- Quick format the SD card



AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

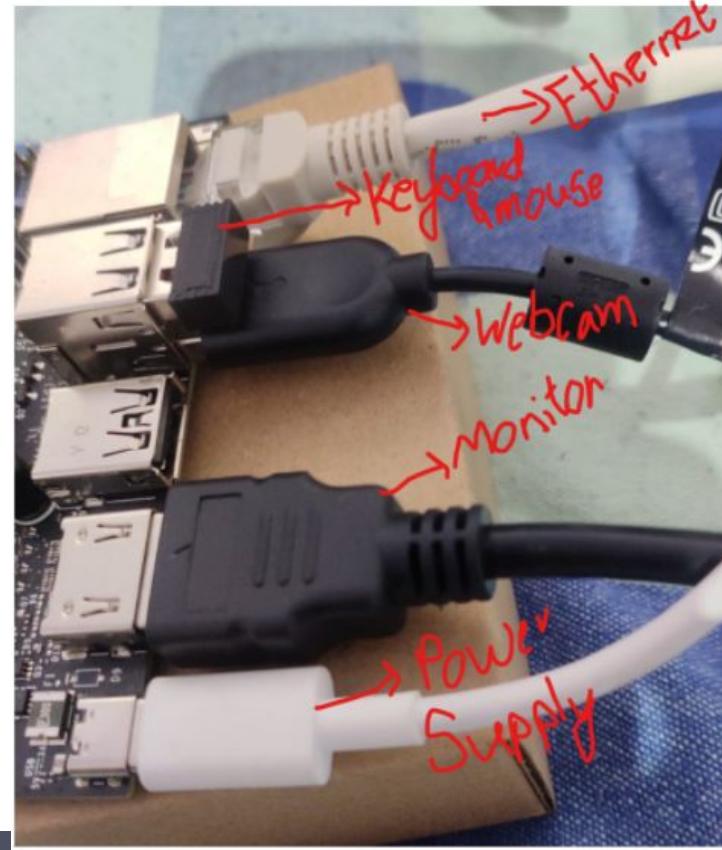
3,) Setting up Kit:-

Insert the SD card in kit



AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

Pasang Aksesoris di slot kit seperti yang ditunjukkan di bawah ini



- Nyalakan catu daya & tunggu sistem untuk boot

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

- Saat Anda boot pertama kali, kit pengembang akan membawa Anda melalui beberapa pengaturan awal, termasuk:
 - Tinjau dan terima perangkat lunak NVIDIA Jetson EULA
 - Pilih bahasa sistem, tata letak keyboard, dan zona waktu
 - Buat nama pengguna, kata sandi, dan nama komputer
 - Konfigurasikan jaringan nirkabel secara opsional
 - Pilih ukuran partisi APP. Disarankan untuk menggunakan ukuran maksimal yang disarankan
 - Buat file swap. Disarankan untuk membuat file swap



AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

4.) Downloading Jetson Inference with Docker Container

Buka Terminal dan ketik perintah berikut

```
git clone --recursive https://github.com/dusty-nv/jetson-inference
```

Tunggu hingga wadah mengunduh (Mungkin membutuhkan waktu 10-15 menit pada koneksi lambat)

Ubah Direktori ke inferensi jetson menggunakan perintah di bawah ini

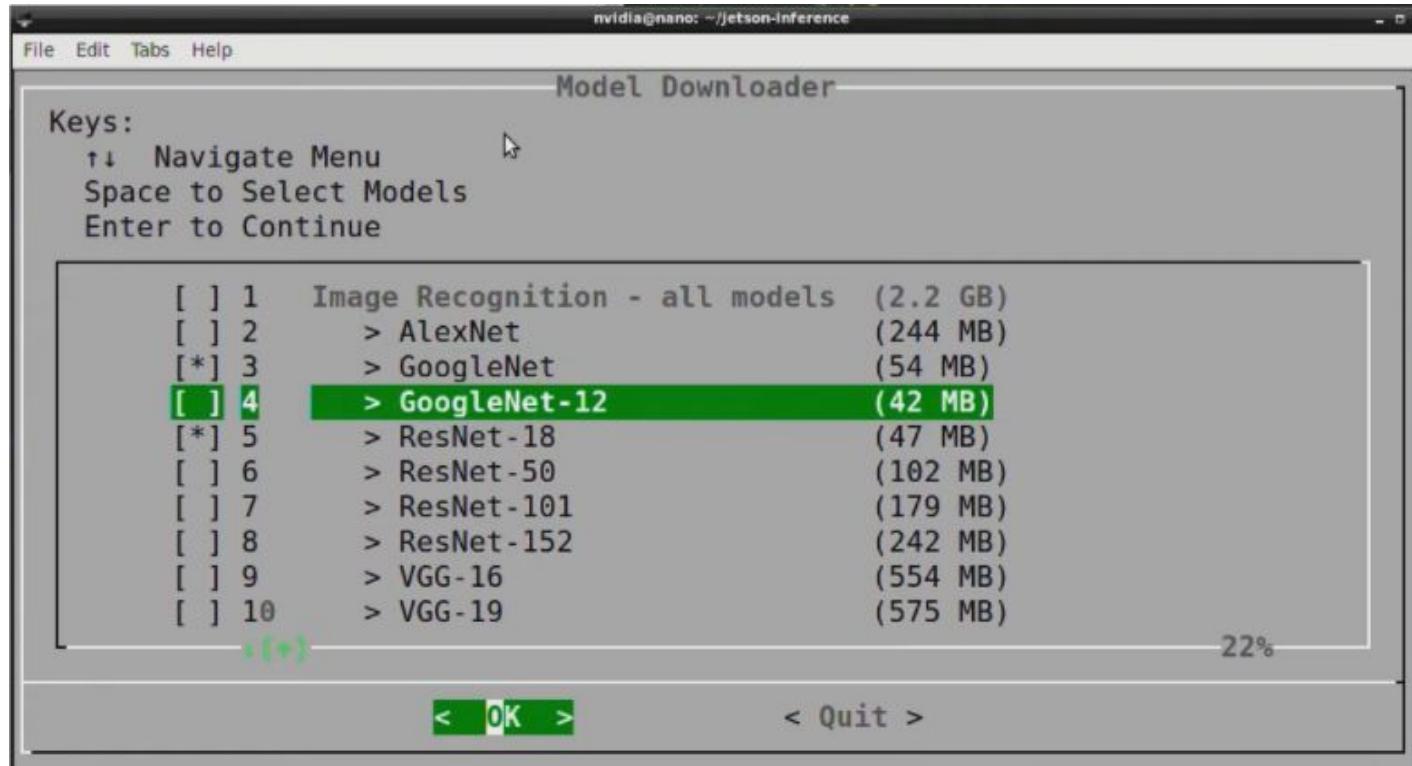
```
cd jetson-inferensi
```

Jalankan perintah wadah Docker. Ini akan meminta kata sandi kit Anda. Masukkan kata sandi (di Linux kata sandi yang Anda ketik tidak ditampilkan) dan tekan enter

```
buruh pelabuhan/run.sh
```

Mungkin perlu waktu dalam Pertama kali menjalankan perintah buruh pelabuhan dan meminta model yang ingin Anda unduh. Unduh model default dan tekan ok

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit



AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

5] Now download the trained dataset from kaggel, we used the following datasets:-

- CT-Scan:-<https://www.kaggle.com/mohamedhanyyy/chest-ctscan-images/download>
- X-ray:-<https://www.kaggle.com/jtiptj/chest-xray-pneumoniacovid19tuberculosis/download>

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

- CT-Scan:-

Ini memiliki Gambar CT-scan paru-paru
Pasien Kanker dan Orang Normal



- rontgen:-

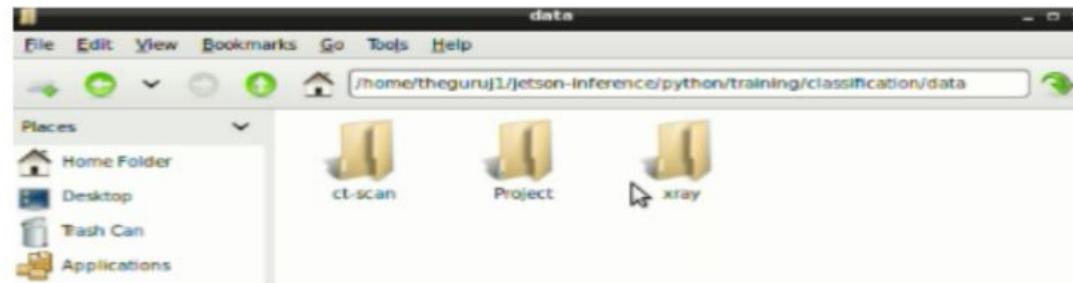
Dataset ini terdiri dari foto rontgen paru pasien Covid-19, pasien Pneumonia, pasien Tuberkulosis dan orang normal.

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

Sekarang Ekstrak dataset di lokasi:

Jetson-inference/python/training/classification/data/

Dengan dataset name. Di folder itu buat file label.txt dan beri nama semua hal yang akan dideteksi oleh kit Anda



AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

6] Melatih Dataset: -

Masukkan perintah berikut

```
python3 train.py --model-dir=models/ct-scan --batch-size=4 --workers=1 --epochs=35 data/ct-scan
```

Ini akan memakan waktu sekitar 8-10 jam untuk pelatihan dan Kit menjadi panas jadi jangan menyentuhnya.

7] Sekarang ekspor model terlatih dalam file onnx

```
python3 onnx_export.py --model-dir=models/ct-scan
```

8] Lakukan hal yang sama dengan dataset lainnya

AI for Healthcare Project using NVIDIA Jetson Nano 2GB Developer kit

9] Sekarang uji proyek: -

```
Imagenet      --model=models/ct-scan/resnet18.onnx      --input_blob=input_0      --output_blob=output_0  
--labels=data/ct-scan/labels.txt (lokasi masukan) (Lokasi keluaran)
```

Untuk deteksi berbasis webcam:

```
Imagenet      --model=models/ct-scan/resnet18.onnx      --input_blob=input_0      --output_blob=output_0  
--labels=data/ct-scan/labels.txt /dev/video0
```

10] Anda dapat Melatih model sekali lagi untuk meningkatkan akurasi atau Silakan coba dengan model yang berbeda untuk membuat proyek yang luar biasa.

Terima
Kasih

